Universidad Carlos iii de Madrid

Escuela Politécnica Superior

Ingeniería Industrial

PROYECTO FIN DE CARRERA

DETECCIÓN DE PERSONAS A PARTIR DE VISIÓN ARTIFICIAL

Autor: Ubaldo González Benítez
Tutor: Dr. Luis Moreno Lorente

Septiembre de 2010

A mis padres, los mejores del mundo.

Agradecimientos

A mi tutor D. Luis Moreno Lorente, tanto por darme la oportunidad de realizar un proyecto tan gratificante, como por su inestimable ayuda a lo largo de estos meses.

A mi familia, de la cual me siento cada día más orgulloso. En especial, de mi hermana Cristina, un ejemplo para crecer a su lado.

A mi compañero de mil batallas Javier Yañéz García, gracias a su ayuda la ingeniería no ha sido para tanto.

A mi novia Kris, por ser la mejor compañera de viaje posible.

A mis amigos de la infancia, por enseñarme el significado de la amistad.

A mis nuevos amigos de la universidad, por lo que hemos vivido, y por lo que queda por vivir.

A los compañeros de laboratorio, en especial a Jorge, Piotr, Alejandro y Fran, por hacer que el tiempo allí fuera mucho más llevadero y por prestarme su ayuda siempre que la necesité.

Al Departamento de Ingeniería de Sistemas y Automática de la Universidad Carlos III, por brindarme las instalaciones para poder desarrollar este proyecto.

Y por último, a todos aquellos profesores que han contribuido en mi formación y han hecho que admire mi nueva profesión, Ingeniero Industrial.

Gracias a todos, de corazón.

Ubaldo González Benítez.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

CAPÍTULO 1

INTRODUCCIÓN

La visión artificial es una rama de la ingeniería electrónica que tiene por objetivo modelar matemáticamente los procesos de percepción visual de los seres vivos y generar programas que permitan simular estas capacidades visuales por ordenador. Es una gran herramienta para establecer la relación entre el mundo tridimensional y las vistas bidimensionales tomadas de él. Por medio de esta teoría se puede hacer, por una parte, una reconstrucción del espacio tridimensional a partir de sus vistas y, por otra parte, llevar a cabo una simulación de una proyección de una escena tridimensional en la posición deseada a un plano bidimensional.

Sus antecedentes se remontan a los años veinte, cuando se mejora la calidad de las imágenes digitalizadas de los periódicos, enviadas por cable submarino entre Londres y Nueva York. Actualmente existen vehículos autónomos que viajan de costa a costa en Estados Unidos y sólo son asistidos por un operador humano el 3 % del tiempo.

El proceso de visión por ordenador puede subdividirse en seis áreas principales:

- 1. Sensado. Es el proceso que nos lleva a la obtención de una imagen visual
- 2. Preprocesamiento. Trata de las técnicas de reducción de ruido y enriquecimiento de detalles en la imagen
- 3. Segmentación. Es el proceso que particiona una imagen en objetos de interés.
- 4. Descripción. Trata con el cómputo de características útiles para diferenciar un tipo de objeto de otro.
- 5. Reconocimiento. Es el proceso que identifica esos objetos.
- 6. Interpretación. Asigna un significado a un conjunto de objetos reconocidos.

Las aplicaciones de la visión artificial en la actualidad son muy variadas e interesantes, a continuación se muestran algunas de ellas:

- Industria automotriz: medición de las dimensiones de cojinetes de frenos, calibración de ensamblado robótico de sensores de frenos 'anti-lock'.
- Industria de dispositivos médicos: inspección de catéteres en el corazón, lectura de códigos en marcapasos.
- Industrias financieras: inspección detallada de tarjetas financieras.
- Retroalimentación visual para robots.
- Comunicación visual hombre-máquina.
- Empresas de seguridad: vídeo-vigilancia.
- Control de tráfico.

En respuesta a la necesidad de los campos mencionados el análisis automático de secuencias de vídeo se ha convertido en un área de investigación muy activa. En este proyecto se estudia en concreto la detección y seguimiento global de personas a través de la visión artificial con el fin de utilizarlo para la retroalimentación visual de robots, mejorando la comunicación e interacción máquina-hombre, o como aplicación de video-vigilancia para empresas de

seguridad. Destacar que con seguimiento global se quiere decir que en caso de haber más de una persona en la escena esta aplicación determina donde se encuentran las personas en cada momento pero sin diferenciar entre ellas, es decir, sin determinar que persona es cada una. Por tanto esta aplicación realiza la detección frame a frame de las personas existentes en el campo de visión pero sin realizar un seguimiento en los frames sucesivos en caso de haber más de una persona, es por ello que a partir de ahora se habla solo de detección si bien con esto se consigue un seguimiento global como se ha comentado.

La detección es una tarea que parece relativamente trivial para los humanos y que es compleja de llevar a cabo por ordenadores.

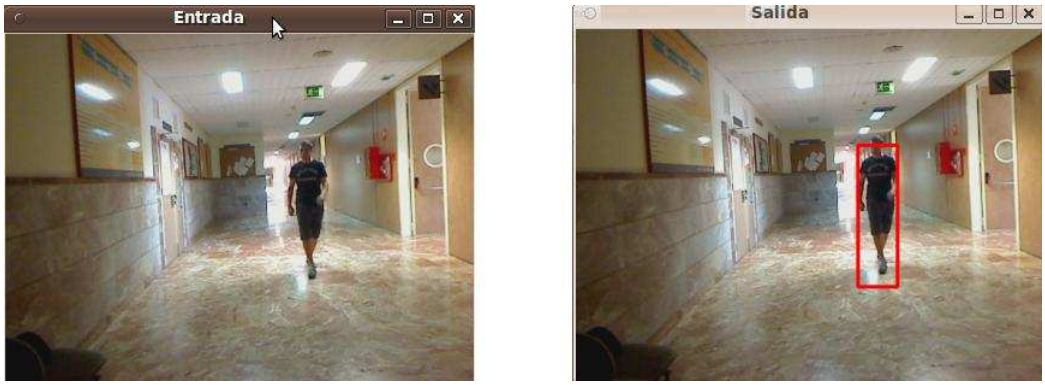

Figura 1.1: Ejemplo de detección de personas.

Las aplicaciones de visión artificial para detección se basan en distinguir entre primer plano en movimiento (del inglés, foreground) y el fondo (del inglés, background). De esta manera se consigue aislar los píxeles en los que hay movimiento del fondo, consiguiendo detectar los objetos que se mueven. Una vez realizado esto se debe agrupar los píxeles en el foreground que conforman un objeto en movimiento para saber su posición, tamaño, etc. Para a continuación distinguir de alguna manera entre personas y el resto de objetos (vehículos por ejemplo).

Una manera efectiva y sencilla de realizar la segmentación FG-BG es extrayendo el background, para lo cual se necesita un modelo preciso y adaptativo de fondo. El background normalmente contiene objetos inertes que permanecen pasivos en la escena (paredes, puertas, muebles...), pero también puede contener objetos no estacionarios. La apariencia de los objetos de background experimenta cambios a lo largo del tiempo como puede ser

debido al cambio de iluminación. Por lo tanto, se puede apreciar que la imagen de background está formada por pixeles estáticos y dinámicos, según se trate de un objeto estacionario o no. Un píxel de background estático puede llegar a convertirse en dinámico con el paso del tiempo y viceversa también, es decir, un píxel de background dinámico puede llegar a transformarse en estático. En definitiva para describir una escena de background general, un modelo de background debe ser capaz de representar la apariencia de un píxel de background estático, un píxel de background dinámico y evolucionar a los cambios tanto repentinos como graduales del background.

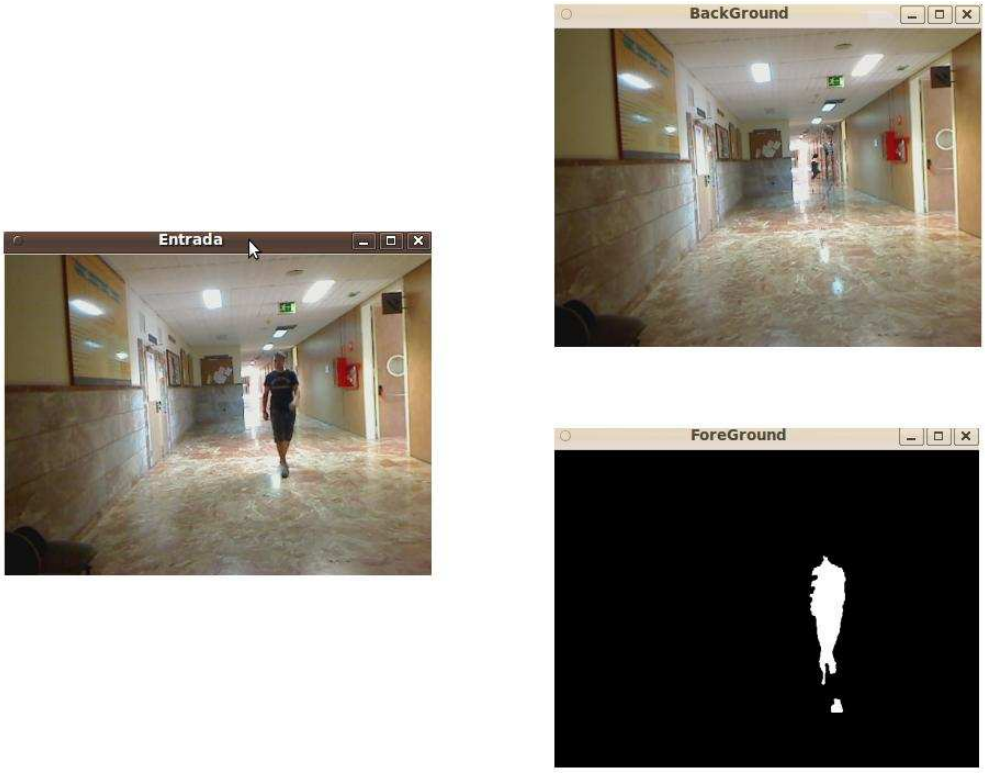

Figura 1.2: Segmentación de imagen en background y foreground.

Por lo tanto, el objetivo principal que persigue el Proyecto Final de Carrera descrito en este documento es el desarrollo de un algoritmo para la detección de personas que se encuentren en el campo de visión a través de la segmentación entre el foreground y el background. Los subobjetivos vinculados al principal son los siguientes:

1. Estudio teórico de la bibliografía existente sobre técnicas de detección con el objetivo de optimizar el algoritmo.

2. Experimentación con la aplicación desarrollada para determinar su eficiencia en los distintos entornos posibles.

3. Análisis de posibles deficiencias del algoritmo para buscarle solución siempre y cuando sea posible.

4. Obtención de conclusiones y posibles trabajos futuros.

En cuanto a la estructura de este documento, este está compuesto por 7 capítulos. En el siguiente capítulo se realiza un resumen de como se encuentra actualmente el estado del arte de la visión por computador, particularizando en la detección de personas. En el se analizarán las distintas tendencias, facilitando al lector la información necesaria para poder situar el proyecto de manera más precisa. En el tercer capítulo se desarrolla un manual sobre las diferentes herramientas y dispositivos necesarios para poder llevar a cabo el proyecto, así como una guía para su instalación. Seguidamente, en el capítulo 4 (Arquitectura funcional de la aplicación), se describe de manera precisa y detallada todos los módulos que conforman la aplicación así como las relaciones entre ellos. Para ello se desarrollan todas las técnicas utilizadas en la aplicación, analizándolas de forma teórica y explicando su aplicación práctica. Una vez mostrados y analizados los módulos del algoritmo, en el capítulo 5 se realizará una serie de experimentaciones, en distintos entornos y ante distintas situaciones, para así determinar el rango de acción y eficacia del mismo. En los dos últimos capítulos, conclusiones y trabajos futuros, respectivamente, se analizan todas las conclusiones obtenidas durante el desarrollo de esta aplicación y se exponen tanto, los puntos donde se debe de seguir mejorando en este campo, como los puntos concretos en los que se puede mejorar este algoritmo.

CAPÍTULO 2

ESTADO DEL ARTE

Como otras tecnologías, la detección mediante visión por ordenador surge de manera gradual, aparece después de un período de investigación y desarrollo en el campo industrial, militar y académico. La aparición de esta tecnología llega acompañada por la madurez en otras tecnologías como los ordenadores, las cámaras digitales y la fibra óptica.

Analizando trabajos anteriores, se observa que uno de los principales requerimientos a la hora de la detección de personas es la localización de las zonas donde se produce el movimiento. Para la detección de dichas regiones, como se comenta en el capítulo 1, se suele utilizar lo que se conoce como técnicas de extracción del background. Destacar que las distintas técnicas existentes se diferencian entre sí solamente en el tipo de modelo de fondo que utilizan y en las técnicas basadas en dicho modelo de fondo para encontrar las regiones estáticas.

A continuación se clasifican los diferentes métodos de segmentación entre regiones estáticas y zonas con movimiento basados en las técnicas de extracción del background. Para que la clasificación sea los más clara posible se ha dividido las técnicas posibles en dos categorías, por un lado las aproxima-

ciones que usan un modelo de fondo y por otro lado las aproximaciones que usan varios modelos de fondo.

El primero de estos grupos se puede dividir, a su vez, en 2 subcategorías dependiendo del uso que dichas aproximaciones hagan con las máscaras de foreground obtenidas:

- Análisis imagen a imagen. Esta categoría describe los métodos que emplean modelos de segmentación frente-fondo bastante comunes, seguidos de otro tipo de análisis. Dependiendo del tipo análisis pueden aparecer varias categorías:
  - Aproximaciones clásicas: basadas en el uso de técnicas sencillas de segmentación frente-fondo y un postprocesado de la máscara de foreground seguido a su vez de alguna otra etapa de análisis.
  - Basados en la acumulación de máscaras de foreground. Dicha acumulación se realiza frame a frame y con ella se puede moldear una máscara final de foreground, de donde se obtienen las regiones estáticas.
  - Basados en algunas propiedades del modelo de fondo utilizado, como por ejemplo considerando las transiciones entre los diferentes estados de un modelo de mezcla de Gaussianas o observando el valor de algunos parámetros como, por ejemplo, el peso de las Gaussianas.
- Análisis de máscaras de foreground muestreadas. Estas aproximaciones intentan detectar regiones estáticas analizando la secuencia de vídeo a diferentes velocidades, aprovechándose de las ventajas espacio-temporales que ello conlleva.

En cuanto al segundo grupo, es decir, aquellas aproximaciones que combinan más de un modelo de fondo para cada píxel, cabe resaltar que son menos utilizadas por los investigadores para tratar de detectar regiones estáticas. Sin embargo, en función de la tasa binaria de procesamiento del vídeo, o del número de modelos de fondo utilizadas para detectar regiones estáticas, se puede hacer la siguiente clasificación:

- Aproximaciones basadas en el análisis imagen a imagen. Se combina las propiedades de los diferentes modelos de fondo que se utilizan.

- Aproximaciones basadas en el sub-muestreo. Estas aproximaciones detectan regiones estacionarias analizando la secuencia de vídeo a través de los diferentes modelos de fondo debido a que cada modelo de fondo se muestrea con una tasa binaria diferente.

En la siguiente figura se muestra un esquema donde se ve de forma clara y concisa toda la clasificación relatada anteriormente.

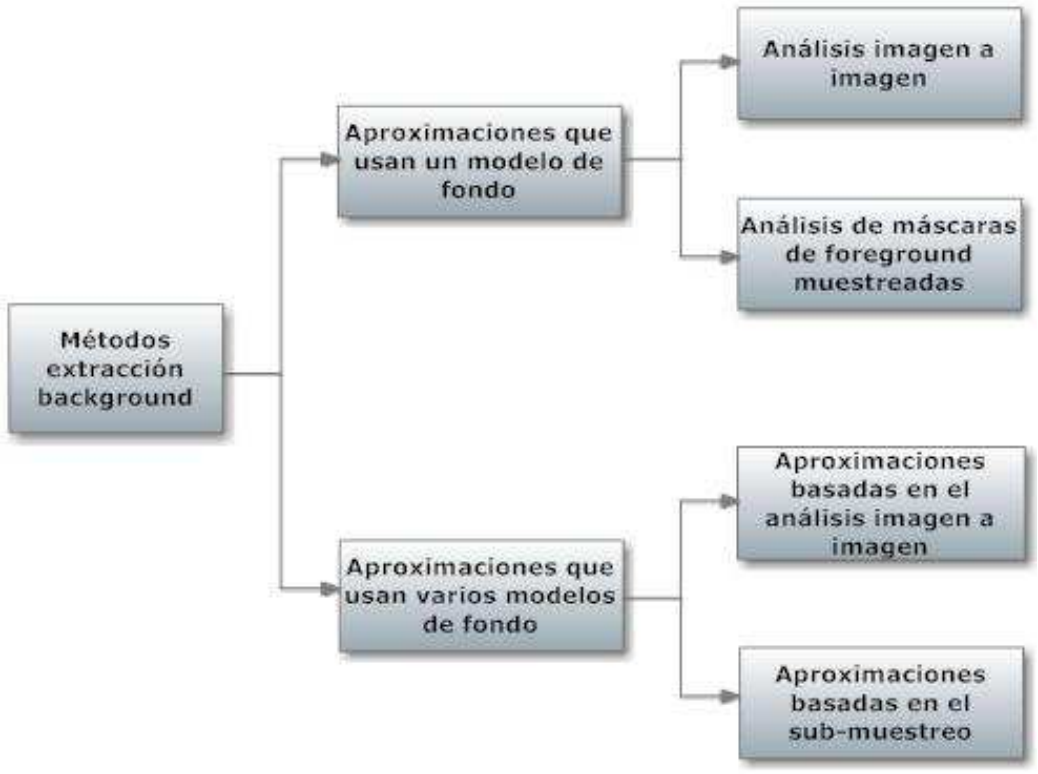

Figura 2.1: Clasificación de métodos de extracción del background.

Una vez se realiza la clasificación general anterior, cabe destacar que los algoritmos más importantes y utilizados para la detección de personas en la actualidad son el modelo de mezcla de Gaussianas (del inglés Mixture of Gaussians: MOG) y el modelo de Bayes (del inglés Foreground Detection based on background modeling and Bayes classification: FGD). El MOG es un modelo de caracterización de los píxeles del fondo basados en el método de mezcla de Gaussianas. Se caracteriza porque tiene en cuenta a la hora de

modelar el fondo los posibles cambios de iluminación en la imagen, secuencias multimodales, objetos moviéndose lentamente, y el ruido introducido por la cámara. Este modelo se utiliza en [4][5][7]. En cuanto al FGD propone un marco bayesiano para incorporar características espectrales, espaciales y temporales en el modelado de background. Deriva una nueva fórmula de la regla de decisión de Bayes para la clasificación de background y foreground. El background es representado usando estadísticas de las principales características asociadas con objetos de background estacionarios y no estacionarios. Se propone un método nuevo para aprender y actualizar las características de background a cambios graduales y repentinos de background. Este modelo se utiliza en [3][15]. Hay que indicar que en el capítulo 3 se realiza un análisis más minucioso y detallado de estos dos últimos algoritmos.

Todas las técnicas anteriormente mencionadas tienen el requisito de que deben ser capaces de detectar objetos en movimiento mediante el uso de un bajo coste computacional y en tiempo real. Además todos los algoritmos deben solventar problemas como:

- Ruido: Es importante que el algoritmo sea capaz de eliminar el ruido procedente de la cámara de vídeo pues puede provocar la detección de zonas de foreground incorrectas. Este ruido es debido al sensor de la cámara o al medio de transmisión de la señal, que se manifiesta en píxeles aislados que toman un valor diferente al de sus vecinos.

- Sombras y reflejos: Es uno de los principales problemas ya que no pertenecen ni al background ni al foreground y en la mayoría de los casos generan interferencias que hacen que el algoritmo no actúe de forma adecuada, por ello hay que eliminarlos. Esto se realiza mediante técnicas de filtrado y operaciones morfológicas.

- Cambios de iluminación: Son variaciones de la iluminación, estas variaciones pueden tener lugar tanto si la escena ha sido capturada en el exterior (cambio de luminosidad a lo lardo del día) como si ha sido tomada en el interior (distintas fuente de iluminación). La iluminación juega un papel importante en la visión artificial pues simplifica de manera considerable el análisis y el procesado de la escena captada. Es un factor que influye en gran medida en la complejidad del algoritmo, y como es más sencillo modificar la iluminación que un complejo algoritmo, hay que dedicar el tiempo necesario para conseguir una buena iluminación de forma que el algoritmo no se complique más de lo debidamente necesario.

- Actualización del fondo de escena: Su importancia radica principalmente en dos puntos. El primero de ellos es que la inicialización del fondo de la secuencia, generalmente no coincide con el background ya que puede haber algún objeto o persona en movimiento, por lo que el algoritmo debe ser capaz de no clasificarlo como fondo, el segundo punto por lo que la actualización de fondo es importante es debido a que pueden haber cambios importantes en la secuencia.

- Determinar los parámetros de funcionamiento de los algoritmos: Es una de las tareas más complejas ya que dependiendo de los parámetros utilizados los algoritmos mostrarán unos resultados u otros, que en la mayoría de los casos difieren en gran medida.

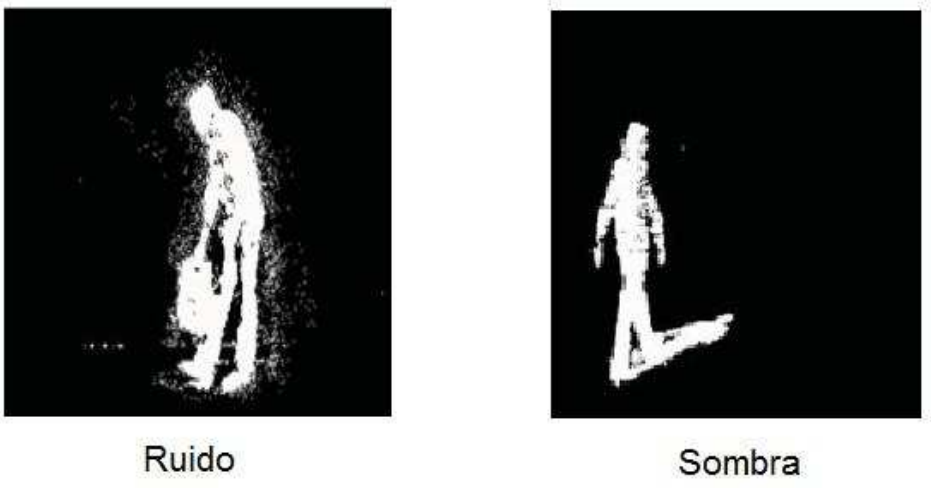

Figura 2.2: Ejemplo de problemas a solventar por el algoritmo de segmentación.

A parte del requerimiento ya analizado de localizar las zonas donde se produce movimiento, el otro gran punto en el que se trata de mejorar los algoritmos de detección de personas en la actualidad es la clasificación automática entre que objetos en movimiento son personas y cuales no. Este es un proceso complejo, debido a que no hay ningún modo fácil de definir la manera en que una persona entra en la escena, ya que las personas pueden adoptar un número indefinido de posturas y además ir modificándola a medida que avanza en la escena. Otro punto que hace que la complejidad de este

proceso aumente, es que las secuencias de vídeo deber ser analizadas en tiempo real, lo que supone que el coste computacional no puede ser demasiado alto.

El algoritmo de clasificación destinado a determinar si un objeto es persona o no se denomina detector de personas. Estos detectores pueden ser de distintos tipos. Pueden ser basados en el análisis de contornos, los cuales son muy convenientes cuando la posturas de las personas a detectar siempre es la misma (siempre andando por ejemplo) y el rango de tipos de personas a detectar es bajo. Otro tipo son los basados en el análisis de regiones, calculando de modo iterativo la elipse más grande contenida en cada región obtenida después de la etapa de segmentación. Por último otro tipo son los basados en las características intrínsecas de la persona(basados en el área o en las relaciones entre las medidas medias de un humano).

Figura 2.3: Detección únicamente de personas.

Finalmente, en cuanto a la dirección que deben tomar las investigaciones futuras en esta área, éstas deberían estar enfocadas a lograr una mayor estabilidad, una menor sensibilidad al ruido y a perfeccionar la detección de personas en ciertos casos puntuales. Todo esto debería permitir desarrollar algoritmos más rápidos, precisos y sencillos, es decir, con un mejor rendimiento que los actuales.

CAPÍTULO 3

ARQUITECTURA HARDWARE Y SOFTWARE

Este capítulo nace con la finalidad de proporcionar al lector toda la información que necesita sobre las distintas herramientas y dispositivos necesarios para ejecutar la aplicación sobre detección que aquí se trata. A continuación se muestran los diferentes puntos a tener en cuenta:

3.1. Sistema operativo

La aplicación se desarrolla sobre el sistema operativo Ubuntu. Este es una distribución Linux basada en Debian GNU/Linux que proporciona un sistema operativo actualizado y estable para el usuario, con un fuerte enfoque en la facilidad de uso y de instalación del sistema. Al igual que otras distribuciones se compone de múltiples paquetes de software normalmente distribuidos bajo una licencia libre o de código abierto.

Una de las grandes ventajas que proporciona este sistema operativo es

que se puede utilizar como sistema de programación permitiendo compilar C, C++, Java, Ada, entre otros muchos lenguajes. Este proyecto es programado bajo el lenguaje de programación C++.

1) Requisitos

Los requisitos mínimos recomendados para ejecutar Ubuntu son los siguientes:

- Procesador: 1 GHz x86.
- Memoria RAM: 512 MB.
- Disco Duro: 5 GB (para una instalación completa con swap incluida).
- Tarjeta gráfica VGA y monitor capaz de soportar una resolución de 1024x768.
- Lector de CD-ROM o tarjeta de red.
- Tarjeta de sonido.
- Conexión a Internet.

Cabe destecar que por lo general se puede ejecutar Ubuntu en hardware más antiguos de lo especificado, aunque el rendimiento necesariamente va a ser menor.

2) Instalación

En caso de no tener instalado el Ubuntu es necesario seguir una serie de sencillos pasos:

1. Descargar el CD de instalación de Ubuntu, el Desktop Cd.
2. El archivo descargado es una imagen ISO que se debe grabar en un disco para proceder con la instalación.
3. Arrancar ordenador desde el CD, para ello reiniciar equipo con el disco grabado en el lector.

4. Por último se deben ir siguiendo los pasos de la instalación. Como ayuda se recomienda visitar la siguiente página web:

   http://www.guia-ubuntu.org/index.php?title=Instalaci%C3%B3n_est%C3%A1ndar

   En ella, a parte de ver los pasos a seguir durante la instalación, también se puede encontrar el Desktop Cd y tutoriales.

3.2. Biblioteca OpenCV 2.1.0

Para la realización del proyecto es necesario descargar e instalar una librería específica, a parte de las bibliotecas generales que vienen incorporadas ya en Ubuntu. El nombre de esta librería es OpenCV 2.1.0.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que aparece su primera versión alfa en el mes de enero de 1999, esta se utiliza en infinidad de aplicaciones. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Como meta el proyecto pretende proveer un marco de desarrollo fácil de utilizar y altamente eficiente. Esto se logra realizando su programación en código c y c++ optimizados, aprovechando además las capacidades que proveen los procesadores multi núcleo. Open CV puede además utilizar el sistema de las primitivas de rendimiento integradas de Intel, que es un conjunto de rutinas de bajo nivel específicas para procesadores Intel.

La web oficial del proyecto es la siguiente:

http://sourceforge.net/projects/opencv/

1) Instalación

Para la instalación de OpenCV 2.1.0 en Ubuntu se deben seguir los siguientes pasos:

1. Se descarga la biblioteca comprimida de la web oficial.
2. Se extrae.
3. Se accede a la carpeta en el terminal.

Figura 3.1: Paso 3 de la instalación de OpenCV 2.1.0

4. Al no tener makefile se descarga el cmake:

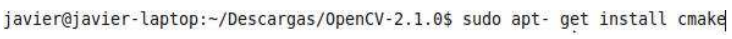

Figura 3.2: Paso 4 de la instalación de OpenCV 2.1.0

5. Se ejecuta el cmake para crear el makefile:

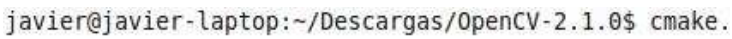

Figura 3.3: Paso 5 de la instalación de OpenCV 2.1.0

6. Se ejecuta el makefile:

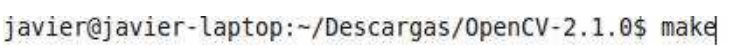

Figura 3.4: Paso 6 de la instalación de OpenCV 2.1.0

7. Por último se instala:

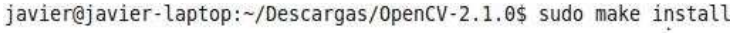

Figura 3.5: Paso 7 de la instalación de OpenCV 2.1.0

Una vez que se ha concluido la instalación se debe tener cuidado con la ruta donde se ha instalado las librerías de OpenCV para cuando se llamen en el programa se referencie la ruta correctamente. En caso de que la posición no sea la que se quiere solo hace falta copiarlas en la ruta elegida mediante la instrucción cp.

3.3. Librería cvblobslib

A parte de las librerías que ya vienen incorporadas al descargar la biblioteca OpenCV 2.1.0, esta aplicación requiere la descarga e instalación de una librería adicional para OpenCV llamada Cvblobslib. Es un componente algo similar a la librería regionprops de Matlab.

1) Instalación

Para la instalación de esta librería se debe, al igual que con la biblioteca OpenCV 2.1.0, descargar la librería de la web oficial. Para encontrar la librería es tan fácil como escribir cvblobslib en el buscador de la web o entrar directamente a la siguiente dirección:

http://opencv.willowgarage.com/wiki/cvBlobsLib

Una vez descargada la librería se deben seguir los mismos pasos que para la instalación de la biblioteca, teniendo en cuenta que lo más recomendable es situar la librería dentro de los include de OpenCV 2.1.0.

3.4. Cámara: Logitech QuickCam Pro 9000

El éxito o no de un algoritmo de visión por ordenador depende en gran medida de la calidad de la imagen sobre la que se trabaja, llegando incluso a ser más importante para el éxito de la aplicación que el propio algoritmo. Es por ello que en este proyecto se ha trabajado con una cámara de vídeo de contrastada válidez suministrada por el departamento de Automática y Robótica de de la Universidad Carlos III, la Logitech QuickCam Pro 9000.

Esta cámara es capaz de producir un vídeo fluido y natural e instantáneas de hasta 8 megapíxeles. Y gracias a su enfoque automático de gama alta las imágenes son siempre nítidas, incluso en primeros planos (a 10 cm de la lente).

1) Especificaciones

- Óptica Zeiss® con enfoque automático.

- Sensor nativo de alta resolución de 2 megapíxeles.
- Vídeo en alta definición (hasta 1600 x 1200*).
- Modo de pantalla panorámica de 720p (con sistema recomendado).
- Fotos de hasta 8 megapíxeles (mejoradas desde el sensor de 2 megapíxeles).
- Micrófono con tecnología Logitech RightSound.
- Vídeo de hasta 30 cuadros por segundo.
- Certificación USB 2.0 de alta velocidad.
- Clip universal para monitores LCD, CRT o portátiles

A continuación se muestra una imagen de la cámara utilizada para la obtención de imágenes en tiempo real.

Figura 3.6: Webcam Logitech QuickCam Pro 9000.

2. Instalación

Al trabajar en el sistema operativo Ubuntu no es necesario realizar ningún tipo de instalación, únicamente conectando la cámara al puerto USB se instalará de forma automática. Por el contrario cabe destacar que para grabar vídeos con la misma es necesario descargar el programa Cheese a través de la aplicación Synaptic de Ubuntu.

CAPÍTULO 4

ARQUITECTURA FUNCIONAL

A continuación se realiza la descripción de la aplicación que aquí se trata, estudiando todos los módulos que la componen, así como los tipos de datos que intercambian entre ellos. Para facilitar la comprensión de la misma al lector en primer lugar se muestra un diagrama de flujo en el que se representan todas las etapas del algoritmo así como la relación entre ellas. Para a continuación de esto analizar en detalle cada módulo, estudiando tanto la teoría de las técnicas utilizadas como su aplicación concreta en la aplicación.

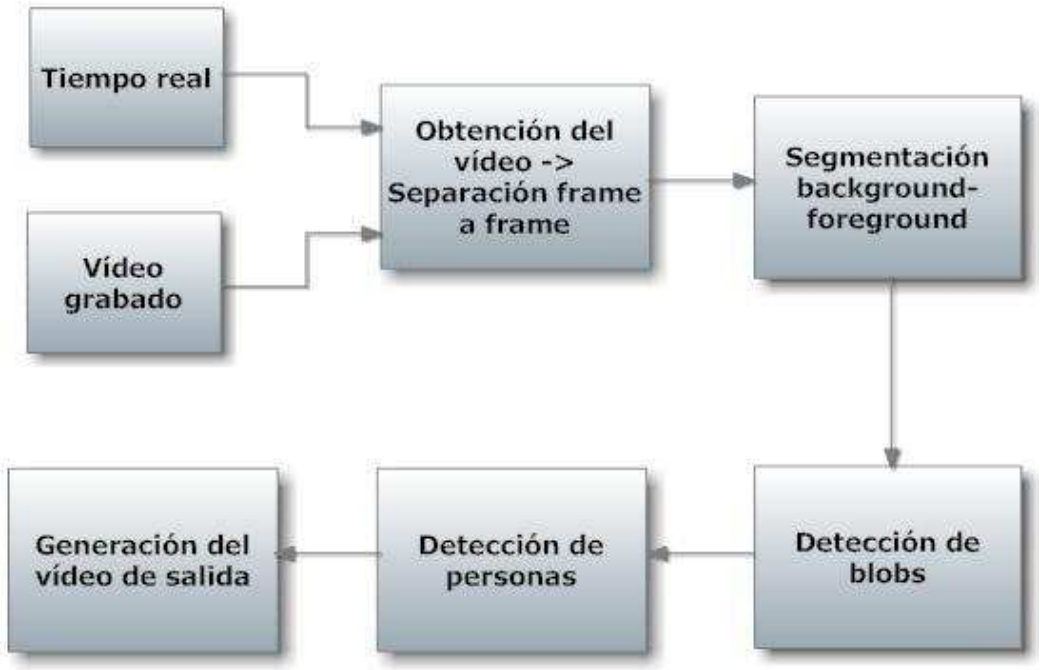

Figura 4.1: Diagrama de flujo de la aplicación.

Antes de comenzar a desglosar cada bloque por separado es necesario definir claramente el concepto de blob, pues es de vital importancia para el entendimiento del proyecto. Un blob es una estructura compuesta por un conjunto de pixeles adyacentes y sus atributos. Este conjunto de píxeles es agrupado tomando en cuenta que cumple con ciertos criterios o parámetros de clasificación y por tanto es tomado como un objeto y no como simples píxeles separados y sin ningún nexo de unión.

4.1. Obtención del vídeo. Separación frame a frame

La obtención del vídeo es el primer módulo de la aplicación y tiene como principal objetivo la adquisición o generación de los datos de imágenes que son usados para su posterior procesamiento en los distintos bloques. Esto se debe a que la aplicación se basa en el procesamiento de imágenes consecutivas, es decir, se divide el vídeo en frames y se trabaja sobre ellos. Para que

sea posible esta división en frames hace falta un proceso el cual se relata a continuación:

Primeramente para generar las imágenes a procesar se utiliza una interfaz gráfica que permite cargar un archivo de vídeo ya existente o capturar una señal de vídeo en vivo, dependiendo si se quiere trabajar en tiempo real o sobre una secuencia ya grabada. Una vez el archivo de vídeo ha sido cargado al módulo, el siguiente paso es capturar cada una de las imágenes que lo componen. Para ello se convierte el formato del vídeo original en un formato que soporte la aplicación, preparado para ser procesado frame a frame. Por último se implementa la división del mismo en imágenes para ir moviéndose por todas y cada una de ellas.

Figura 4.2: División del vídeo de entrada en frames.

4.2. Segmentación de fondo

Como se menciona en el capítulo 1 y 2, uno de los módulos principales en un algoritmo de detección es el de extracción del primer plano en movimiento

del fondo para detectar las zonas en las que hay movimiento. Este debe ser el primer módulo que se debe aplicar a la imagen de entrada del vídeo en tiempo real, tal y como se ve en el diagrama de la aplicación (figura 4.1).

Es evidente que el éxito que se consiga en este bloque es de vital importancia para el válido funcionamiento del resto del algoritmo, pero también es evidente que dado que se trata de un programa con el objetivo de trabajar en tiempo real hay que tener muy en cuenta el coste computacional.

Indicar que a priori la extracción de las zonas en las que hay movimiento en la imagen es más sencilla en entornos de interior que en exteriores debido a que estos últimos se ven más afectados por cambios de iluminación, movimientos de ramas de un árbol o movimientos en las superficies de agua por ejemplo.

A continuación se muestra un diagrama en el que se puede observar como trabaja este bloque independientemente del tipo de modelo concreto que se use en esta aplicación (para ver tendencias actuales en este ámbito ver capítulo 2):

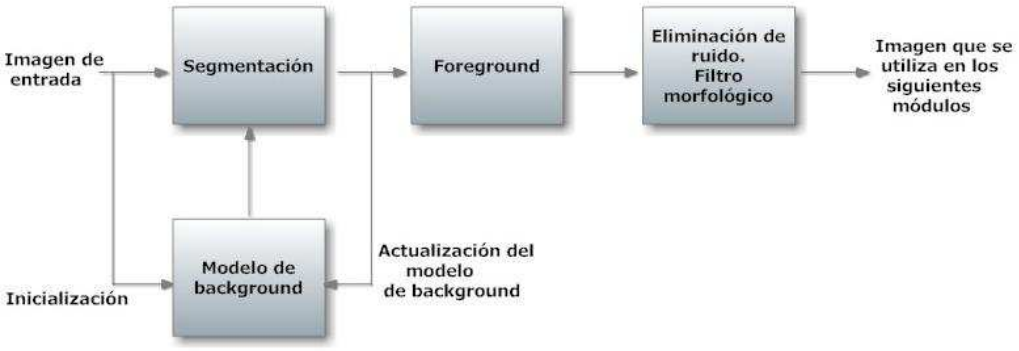

Figura 4.3: Diagrama de flujo de la segmentación de fondo.

Como se puede observar en la figura, después del módulo de segmentación background/foreground (con la realimentación necesaria para ir actualizando el background) se sitúa un filtro morfológico para eliminar el ruido. Cabe indicar que en algún algoritmo de segmentación ya se utiliza un pequeño filtro morfológico, pero aun así se comprueba que los resultados obtenidos son mejores realizando un filtro morfológico a la salida del sub-bloque para obtener así menos ruido.

Seguidamente se analizan por separado estos dos bloques. En un primer punto se estudia los dos algoritmos más utilizados en la actualidad para extraer el foreground que cumplen con una óptima relación eficiencia-coste computacional, para a continuación elegir uno de ellos para su aplicación en el proyecto. Por último se explica el filtro morfológico utilizado para extraer el ruido.

4.2.1. Análisis teórico de los algoritmos para extracción de fondo

Como se indica en el capítulo 2, los dos algoritmos principales a analizar son:

- FGD (Foreground Detection based on background modeling and Bayes classification).
- MOG (Mixture of Gaussians).

A continuación se realiza una explicación teórica de los mismos, intentando que sea lo más clara y amena posible e incluyendo las fórmulas matemáticas necesarias para su fácil comprensión, las cuales son desarrolladas en [3] y [4].

Algoritmo FGD

El background es normalmente representado por características o rasgos de imagen en cada píxel. Las características extraídas de una secuencia de imágenes pueden ser clasificadas en tres tipos: características espectrales, espaciales y temporales. Las características espectrales están asociadas a escala de grises o información de color, las características espaciales están asociadas al gradiente o estructura local, y las características temporales están asociadas a cambios entre frames en el píxel. Las dos primeras son adecuadas para describir la apariencia de los píxeles de background estáticos, mientras que las terceras se usan para describir los píxeles de background dinámicos asociados con los objetos no estacionarios.

Este método o algoritmo propone un marco bayesiano que incorpora múltiples tipos de características para modelar backgrounds complejos. Los puntos importantes del método propuesto son los que siguen:

- 1) Se propone un marco bayesiano para incorporar características espectrales, espaciales y temporales en el modelado de background.
- 2) Se deriva una nueva fórmula de la regla de decisión de Bayes para la clasificación de background y foreground.
- 3) El background es representado usando estadísticas de las principales características asociadas con objetos de background estacionarios y no estacionarios.
- 4) Se propone un novedoso método para el aprendizaje y la actualización de las características de fondo a cambios graduales y repentinos de background
- 5)Se analiza la convergencia del proceso de aprendizaje y se deriva una formúla para seleccionar un ratio de aprendizaje adecuado.
- 6)Se desarrolla un nuevo algoritmo en tiempo real para la detección de objetos de foreground en entornos complejos.

A continuación se desarrolla el algoritmo.

I. MODELADO ESTADÍSTICO DEL FONDO (BACKGROUND)

A. Clasificación de Bayes del background y foreground

Para objetos o regiones de background y foreground arbitrarios, la clasificación del background o foreground puede ser formulada bajo la teoría de decisión de Bayes.

Sea s=(x,y) la posición de un píxel de una imagen, I(s,t) la imagen de entrada en el instante t, y v un vector característico n-dimensional extraído de la posición s en el instante t de la secuencia de la imagen. Entonces, la probabilidad posterior de que el vector característico del background en s puede ser calculada mediante el uso de la regla de Bayes es:

$$P_s(b \mid v) = \frac{P_s(v \mid b)P_s(b)}{P_s(v)} \tag{4.1}$$

donde:

b: background.

$P_s(v|b)$:probabilidad del vector característico v, siendo observado como un background en s.

$P_s(b)$: probabilidad anterior del píxel s perteneciendo al background.

$P_s(v)$: probabilidad anterior del vector característico v siendo observado en la posición s.

Igualmente, la probabilidad posterior de que el vector característico v venga de un objeto de foreground en s es:

$$P_s(f \mid v) = \frac{P_s(v \mid f)P_s(f)}{P_s(v)} \tag{4.2}$$

donde:

f: foreground

Usando la regla de decisión de Bayes, un píxel s es clasificado como perteneciente al background acorde a su vector característico v observado en el instante t si:

$$P_s(b \mid v) > P_s(f \mid v) \tag{4.3}$$

De otra forma, es clasificado como perteneciente al foreground. Hay que tener en cuenta que un vector característico observado en un píxel de imagen viene de objetos de background o de objetos de foreground, esto supone:

$$P_s(v) = P_s(v \mid b)P_s(b) + P_s(v \mid f)P_s(f) \tag{4.4}$$

Sustituyendo (4.1) y (4.4) en (4.3) se concluye que la regla de decisión de Bayes se convierte en:

$$2P_s(v \mid b)P_s(b) > P_s(v) \tag{4.5}$$

B. Representación de la característica principal del background

Para aplicar (4.5) para la clasificación del background y foreground, las funciones de probabilidad Ps(b), Ps(v) y $Ps(v \mid b)$ deben ser conocidas en adelante, o deben poder ser estimadas adecuadamente. Para backgrounds complejos, las formas de estas funciones de probabilidad son desconocidas. Una manera de estimar estas funciones de probabilidad es usando el histograma de características. El problema que se encuentra es el alto coste de almacenamiento y cálculo computacional. Asumiendo que v es un vector n-dimensinal y cada uno de sus elementos es cuantizado a L valores, el histograma contendría L^n celdas. Por ejemplo, asumiendo que la resolución de color tiene 256 niveles, el histograma contendría 256^3 celdas. El método sería irreal en términos de requerimientos computacionales y de memoria.

Es razonable asumir que si las características seleccionadas representan el background efectivamente, el incremento de las características de background debería ser pequeño, lo que implica que la distribución de características de background estará altamente concentrada en una pequeña región del histograma. Además, las características de varios objetos de foreground se propagarían ampliamente en el espacio característico. Esto implica que, con una adecuada selección y cuantización de características, sería posible describir aproximadamente el background usando solamente un número pequeño de vectores característicos. Una estructura de información concisa para implementar tal representación de background es creada como sigue.

Sean v_i los vectores característicos cuantizados clasificados en orden descendente con respecto a $P_s(v_i \mid b)$ por cada píxel s. Entonces, para una selección adecuada de características, habría un pequeño entero N(v), un alto porcentaje de M_1, y un valor de porcentaje bajo de M_2 (por ejemplo M_1=80 %-90 % y M_2=10 %-20 %) tales que el background estaría aproximado por:

$$\sum_{i=1}^{N(v)} P_s(v_i \mid b) > M_1 \text{ y } \sum_{i=1}^{N(v)} P_s(v_i \mid f) < M_2 \tag{4.6}$$

El valor de N(v) y la existencia de M_1 y M_2 dependen de la selección y cuantización de los vectores característicos. Los N(v) *vectores característicos* son definidos como las principales características del background en el píxel s.

Para aprender y actualizar las probabilidades anterior y condicional para los vectores característicos principales, se establece una tabla de estadísticas para las características principales posibles para cada tipo de característica en s. La tabla se denota como:

$$T_v(s) = \begin{cases} p_v^t(b) \\ \{S_v^t(i)\},\ i = 1, ..., M(v) \end{cases} \tag{4.7}$$

donde:

$p_v^t(b)$: es el $P_s(b)$ aprendido, basado en la observación de las características v.

$S_v^t(i)$ registra las estadísticas de los M(v) vectores característicos más frecuentes (M(v)>N(v)) en el píxel s.

Cada $S_v^t(i)$ contiene tres componentes:

$$S_v^t(i) = \begin{cases} p_{vi}^t = P_s(v_i) \\ p_{vi|b}^t = P_s(v_i \mid b) \\ v_i = (v_{i1}...v_{iD(v)})^T \end{cases} \tag{4.8}$$

donde:

D(v): dimensión del vector característico v.

Los $S_v^t(i)$ en la tabla $T_v(s)$ son clasificados en orden descendente con respecto al valor p_{vi}^t. Los primeros elementos N(v) de la tabla $T_v(s)$, junto con $p_v^t(b)$, son usados en (4.5) para la clasificación de background y foreground.

C. Selección de característica

La siguiente cuestión esencial para la representación de la característica principal es la selección de característica. Las características significantes de distintos objetos de background son diferentes. Para lograr la representación eficaz y precisa de los píxeles de background con características principales, el empleo de tipos adecuados de características es importante. Tres tipos de características, espectrales, espaciales y temporales, son usadas para el modelado de background

a) Características para píxeles de background estáticos: Para un píxel perteneciente a un objeto de background estacionario, las características más importantes y estables son su color y estructura local(gradiente). Por lo tanto, se usan dos tablas para aprender las características principales. Estas son $Tc(s)$ y $Te(s)$ con $c = [R, G, B]^T$ y $e = [gx, gv]^T$ representando los vectores de color y gradiente, respectivamente. Debido a que el gradiente es menos sensitivo a cambios de iluminación, los dos tipos de vectores característicos pueden ser integrados bajo el marco de Bayes como se muestra a continuación.

Sea $v = [c^T, e^T]^T$ y se asume que c y e son independientes, la regla de decisión de Bayes (4.5) se convierte en :

$$2P_s(c \mid b)P_s(e \mid b)P_s(b) > P_s(c)P_s(e) \tag{4.9}$$

Para las características de los píxeles de background estáticos, la medida de cuantización debería ser menos sensitiva a cambios de iluminación. Aquí, una medida de distancia normalizada basada en el producto interno de dos vectores es empleada por ambos vectores de color y gradiente. La medida de distancia es:

$$d(v_1, v_2) = \frac{1 - 2(v_1, v_2)}{||v_1||^2 + ||v_2||^2} \tag{4.10}$$

donde v puede ser c o e, respectivamente.

Si $d(v_1, v_2)$ es menor que un pequeño valor σ, v_1 y v_2 están emparejados entre si. La robustez de la medida de distancia (4.10) a los cambios de iluminación y las imágenes de ruido se muestra en [19]. El vector de color C es obtenido directamente de las imágenes de entrada con 256 niveles de resolución para cada componente, mientras que el vector de gradiente e es obtenido mediante la aplicación de operador Sobel para las correspondientes imágenes de entrada de escala de grises con 256 niveles de resolución. Con σ=0.005, N(v)=15 se encuentra la suficiente precisión para aprender las características principales para los píxeles de background estáticos.

Un ejemplo de la principales características representadas para un píxel de background estático se muestra en la figura (4.4), donde se muestran los histogramas para las características de color y gradiente principales en $T_c(s)$

y $T_e(s)$. El histograma de las características de color muestra que sólo los 2 primeros son colores principales para el background, y el histograma de gradientes muestra que los 6 primeros, excluyendo el cuarto, son los principales gradientes para el background.

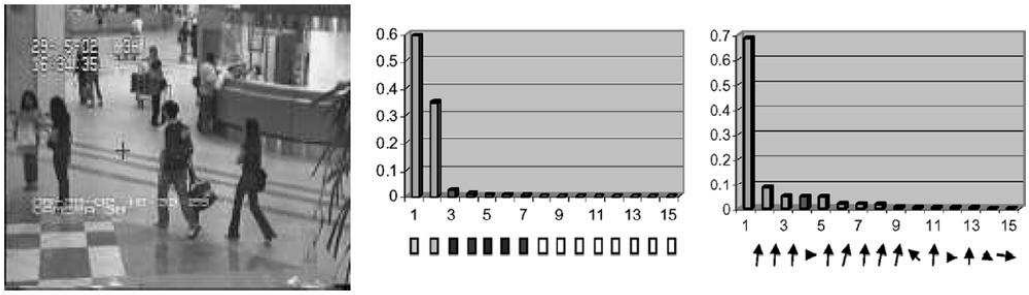

Figura 4.4: Ejemplo de aprendizaje de las principales características para un píxel de backgrpund estático en una escena concurrida. La imagen de la izquierda muestra la posición del píxel seleccionado. Las dos imágenes de la derecha son los histogramas de los estadísticos de color y gradiente más importantes, donde la altura de una barra es p^t_{vi}, la parte de luz de gris es $p^t_{vi|t}$ y el máximo de la parte de oscuridad de gris es $p_v i^t$-$p^t_{vi|t}$. Los iconos que se muestran debajo de los histogramas corresponden a las características de color y gradiente.

b) Características para píxeles de background dinámicos: Para píxeles de background dinámicos asociados con objeto no estacionarios, las co-ocurrencias de color son usadas como sus características dinámicas. Esto es porque la co-ocurrencia de color entre frames consecutivos es considerada adecuada para describir las características dinámicas asociadas con objetos de background no estacionarios. Dando un cambio entre frames del color $c_{(t-1)} = [R_{t-1}, G_{t-1}, B_{t-1}]^T$ a $c_t = [R_t, G_t, B_t]^T$ en el instante de tiempo t y el píxel s($c_t \neq c_{(t-1)}$), el vector característico de co-ocurrencia de color es definido como $v = cc = [R_{t-1}, G_{t-1}, B_{t-1}, R_t, G_t, B_t]^T$. De manera similar, una tabla de estadísticas para co-ocurrencia de color $T_{cc}(s)$ es mantenida en cada píxel. Sea $I(s,t) = [I_R(s,t), I_G(s,t), I_B(s,t)]^T$ la imagen de color entrante; el vector de co-ocurrencia de color cc es generado por medio de la cuantización de los componentes de color a baja resolución. Por ejemplo, mediante la cuantización de la resolución de color a 32 niveles para cada componente y seleccionado N(cc) =50, se obtendría una representación característica principal buena para un píxel de background dinámico. Un ejemplo de la característica principal de representación con co-concurrencia de color para una imagen parpadeante

se muestra en la figura (4.5).Comparado con el espacio característico de co-ocurrencia de color de 32^6 celdas, $N(cc) = 50$ implica que con un número muy pequeño de vectores característicos, las características principales son capaces de modelar los píxeles de background dinámicos.

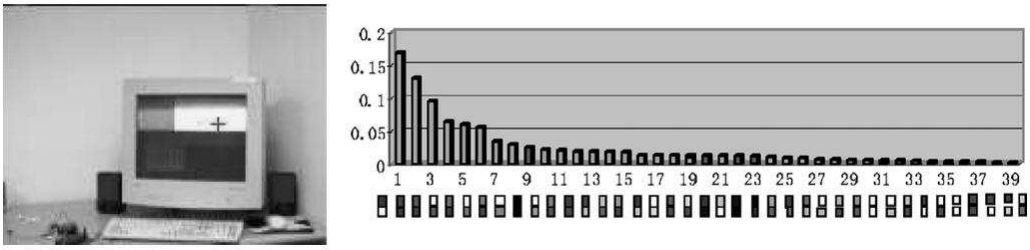

Figura 4.5: Ejemplo de aprendizaje de las principales características para un píxel de background dinámico. La imagen de la izquierda muestra la posición del píxel seleccionado. La imagen de la derecha es el histograma de los estadísticos para las más importantes co-ocurrencias de color en $T_{cc}(s)$, donde la altura de una de sus barras es el valor de p^t_{vi}, la parte de luz de gris es $p^t_{vi|t}$ y el máximo de la parte de oscuridad de gris is $p_v i^t$-$p^t_{vi|t}$. Los iconos que se muestran debajo del histograma corresponden a las características de co-ocurrencia de color y gradiente. En la imagen el color cambia en blanco, azul oscuro y azul claro periódicamente.

II. Aprendizaje y actualización de las estadísticas para las características principales

Dado que el marco background puede sufrir tanto cambios graduales como repentinos, se proponen dos estrategias para aprender y actualizar las estadísticas para las características principales.

A. Para cambios de background graduales

En cada instante de tiempo, si el píxel s es identificado como un punto estático, las características de color c y gradiente e son usadas para su clasificación como foreground o background. De otra manera, se usa la característica de co-ocurrencia de color cc. Asumamos que el vector característico v es usado para clasificar el píxel s en el tiempo t basándose en características principales aprendidas previamente. Entonces las estadísticas de los vectores característicos correspondientes en la tabla $T_v(s)$ (v=c y e, o cc) son gradualmente actualizadas en cada instante de tiempo por:

$$\begin{aligned} p_v^{t+1}(b) &= (1-\alpha)p_v^t(b) + \alpha L_b^t \\ p_{vi}^{t+1} &= (1-\alpha)p_{vi}^t + \alpha L_{vi}^t \\ p_{vi|b}^{t+1} &= (1-\alpha)p_{vi|b}^t + \alpha(L_b^t L_{vi}^t) \end{aligned} \tag{4.11}$$

donde:

El ratio de aprendizaje α es un número positivo pequeño.

Y i=1,...,M(v).

En (4.11) $L_b^t = 1$ significa que s es clasificado como un punto de background en el tiempo t en la segmentación final, de otra manera, $L_b^t = 0$. De manera similar $L_{vi}^t = 1$ significa que el vector ith de la tabla $T_v(s)$ concuerda con el vector característico v, y de otra manera $L_{vi}^t = 0$.

De la operación de actualización anterior se deduce lo siguiente. Si el píxel s es etiquetado como un punto de background en el tiempo t, $p_v^{t+1}(b)$ es incrementado ligeramente respecto a $p_v^t(b)$ debido a $L_b^t = 1$. Además, las probabilidades para el vector característico casado son también incrementadas debido a $L_{vi}^t = 1$. Sin embargo, si $L_{vi}^t = 0$, entonces las estadísticas para los vectores característicos que no casan son decrementadas ligeramente. Si no hay concordancia entre el vector característico v y los vectores en la tabla $T_v(s)$, el vector M(v)th en la tabla es remplazado por un vector característico nuevo.

$$p_{vM(v)}^{t+1} = \alpha,\ p_{vM(v)|b}^{t+1} = \alpha,\ V_{M(v)} = v \tag{4.12}$$

Si el píxel s es etiquetado como un punto de foreground en el tiempo t, $p_v^{t+1}(b)$ y $p_{v_i|b}^{t+1}$ son decrementados ligeramente con $L_b^t = 0$. Sin embargo, el vector casado en la tabla $p_{v_i}^{t+1}$ es incrementado ligeramente.

Los elementos actualizados en la tabla $T_v(s)$ son reclasificados en orden descendente con respecto a p_{vi}^{t+1}, tal que la tabla mantendría los vectores característicos M(v) más frecuentes y significantes observados en el píxel s.

B. Para cambios de background repentinos

Acorde a (4.4), los estadísticos de las características principales satisfacen:

$$\sum_{i=1}^{N(v)} P_s(v_i) = P_s(b) \sum_{i=1}^{N(v)} P_s(v_i \mid b) + P_s(f) \sum_{i=1}^{N(v)} P_s(v_i \mid f) \tag{4.13}$$

Estas probabilidades son aprendidas gradualmente con operaciones descritas por (4.11) y (4.12) en cada píxel s. Cuando se produce un cambio de background repentino, la nueva apariencia del background pronto llega a ser dominante después del cambio. Con la operación respuesta (4.12),con la operación de acumulación gradual (4.11) y reordenando en cada paso de tiempo, las características nuevas aprendidas serán gradualmente movidas a las primeras posiciones pequeñas en $T_v(s)$. Después de un tiempo de duración, el término en la parte izquierda de (13) llega a ser grande (≈ 1) y el primer término en la parte derecha llega a ser muy pequeño debido a que las nuevas características de background son clasificadas como foreground. Una nueva apariencia de background en s puede ser encontrada a partir de (4.6) y (4.13) si:

$$P_s(f) \sum_{i=1}^{N(v)} P_s(v_i \mid f) = \sum_{i=1}^{N(v)} P_s(v_i) - P_s(b) \sum_{i=1}^{N(v)} P_s(v_i \mid b) > M1 \tag{4.14}$$

donde:

b: background anterior al cambio repentino.

f: nueva apariencia del background después del cambio repentino.

El factor $P_s(f)$ previene errores causados por un pequeño número de características de foreground. Usando la notación en (4.7) y (4.8), la condición (4.14) llega a ser:

$$\sum_{i=1}^{N(v)} p_{vi}^t - p_v^t(b) \sum_{i=1}^{N(v)} p_{v_i|b}^t > M1 \tag{4.15}$$

Una vez que la condición anterior está satisfecha, los estadísticos para el foreground son afinadas para ser la nueva apariencia de background. Acorde a (4.4), la operación de aprendizaje repentina es realizada como sigue:

$$\begin{array}{l} p_v^{t+1}(b) = 1 - p_v^t(b) \\ p_{vi}^{t+1} = p_{vi}^t \\ p_{vi|b}^{t+1} = \frac{(p_{vi}^{t+1} - p_v^t(b)_{vi|b}^t)}{p_v^{t+1}(b)} \end{array} \tag{4.16}$$

para i=1...N(v).

C. Convergencia del proceso de aprendizaje

Si la representación de característica principal ha aproximado satisfactoriamente el background, entonces $\sum_{i=1}^{N(v)} p_{v_i|b}^t \approx 1$ debería estar satisfecha. Además, es deseable que $\sum_{i=1}^{N(v)} p_{v_i|b}^t$ converja a 1 con la evolución del proceso de aprendizaje. Se puede mostrar que la operación de aprendizaje (4.11) se encuentra como una condición.

Supongamos $\sum_{i=1}^{N(v)} p_{v_i|b}^t = 1$ en el tiempo t, y el vector jth en la tabla $T_v(s)$ empareja con el vector característico de entrada v que ha sido detectado como background en la segmentación final en el tiempo t. Entonces, de acuerdo a (4.11), se tiene:

$$\sum_{i=1}^{N(v)} p_{vi|b}^{t+1} = (1-\alpha) \sum_{i=1}^{N(v)} p_{vi|b}^{t+1} + \alpha(L_b^t L_{vj}^t) = 1 - \alpha + \alpha = 1 \tag{4.17}$$

Esto implica que la suma de las probabilidades condicionales de las características principales siendo background permanecerá igual o cercana a 1 durante la evolución del proceso de aprendizaje. Supongamos$\sum_{i=1}^{N(v)} p_{v_i|b}^t \neq 1$ en el tiempo t debido a algunas razones tales como el alboroto de objetos de foreground o la operación de aprendizaje repentino, y el v_j, de los primeros vectores N(v) en $T_v(s)$ casan con el vector característico de entrada v, luego tenemos:

$$\sum_{i=1}^{N(v)} p_{vi|b}^{t+1} = (1-\alpha) \sum_{i=1}^{N(v)} p_{vi|b}^{t+1} + \alpha(L_b^t L_{vj}^t) \tag{4.18}$$

Si el píxel s es detectado como un punto de background en el tiempo t, ello conduce a:

$$\sum_{i=1}^{N(v)} p_{vi|b}^{t+1} - \sum_{i=1}^{N(v)} p_{vi|b}^{t} = \alpha(1 - \sum_{i=1}^{N(v)} p_{vi|b}^{t}) \tag{4.19}$$

Si $\sum_{i=1}^{N(v)} p_{v_i|b}^{t} < 1$, entonces $\sum_{i=1}^{N(v)} p_{v_i|b}^{t+1} > \sum_{i=1}^{N(v)} p_{v_i|b}^{t}$. En este caso, la suma de las probabilidades condicionales de las características principales siendo background se incrementa ligeramente. Por otra parte, si $\sum_{i=1}^{N(v)} p_{v_i|b}^{t} < 1$, será $\sum_{i=1}^{N(v)} p_{v_i|b}^{t+1} > \sum_{i=1}^{N(v)} p_{v_i|b}^{t}$, y la suma de las probabilidades condicionales de las características principales siendo background se decrementa ligeramente. De estos dos casos, puede ser concluido que la suma de las probabilidades condicionales de las características principales siendo background converge a 1 mientras las características de background sean observadas frecuentemente.

D. Selección del ratio de aprendizaje

En general, para un filtrado IIR basado en el proceso de aprendizaje, hay una compensación en la selección de la tasa de aprendizaje(α).Para hacer el proceso de aprendizaje adaptable a los cambios de background graduales y no ser perturbado por el ruido y objetos de foreground, un valor pequeño sería seleccionado para α. Por otra parte, si α es demasiado pequeño, el sistema llegaría a ser demasiado lento para responder a los cambios de background repentinos. Aquí, se deriva una fórmula para seleccionar α acorde al tiempo requerido por el sistema para responder a los cambios de background repentinos.

Un cambio de background repentino ideal en el tiempo t_0 puede ser asumido para ser una función de paso. Suponiendo que las características antes de t_0 caen hacia los primeros K_1 vectores en la tabla $T_v(s)$ ($K_1 < N(v)$), y las características después de t_0 caen hacia los siguientes K_2 elementos de $T_v(s)$ ($K_2 < N(v)$). Entonces, las estadísticas en el tiempo t_0 pueden ser descritas como:

$$\begin{array}{l} \sum_{i=1}^{K_1} p_{vi}^{t0} \approx 1,\ \sum_{i=1}^{K_1} p_{vi|b}^{t0} \approx 1,\ p_v^{t0}(b) \approx 1 \\ \sum_{i=K_1+1}^{K_1+k_2} p_{vi}^{t0} \approx 0,\ \sum_{i=k1+1}^{K1+k2} p_{vi|b}^{t0} \approx 0 \end{array} \tag{4.20}$$

Debido a que la nueva apariencia de background en el píxel s después del tiempo t_0 es clasificado como foreground antes de la actualización repentina con (4.16),$p_v^{tk}(b)$,$\sum_{i=1}^{K_1} p_{v_i}^{t_k}$ y $\sum_{i=1}^{K_1} p_{v_i|b}^{t_k}$ decrementan exponencialmente,

mientras que $\sum_{i=k_1+1}^{K_1+K_2} p_{vi}^{t_k}$ incrementa exponencialmente y será cambiado a las primeras K_2 posiciones en la tabla actualizada $T_v(s)$ con clasificación a cada paso de tiempo. Inmediatamente la condición de (4.15) es encontrada en el tiempo t_0, el nuevo estado de background es aprendido. Para hacer la expresión más simple, se asume que no hay operación de reordenación. Entonces la condición (4.15) llega a ser:

$$\sum_{i=K_1+1}^{K_1+k_2} p_{vi}^{tn} - p_v^{tn}(b) \sum_{i=K_1+1}^{K_1+k_2} p_{vi|b}^{tn} > M_1 \tag{4.21}$$

De (4.11) y (4.20), se sigue que en el tiempo t_n, las condiciones siguientes sostienen:

$$p_v^{tn}(b) = (1-\alpha)^n p_v^{t0}(b) = (1-\alpha)^n \tag{4.22}$$

$$\sum_{i=K_1+1}^{K_1+k_2} p_{vi}^{tn} = (1-\alpha)^n \sum_{i=K_1+1}^{K_1+k_2} p_{vi}^{t0} + \sum_{j=0}^{n-1}(1-\alpha)^j \alpha \approx 1-(1-\alpha)^n \tag{4.23}$$

$$\sum_{i=K_1+1}^{K_1+k_2} p_{vi|b}^{tn} = (1-\alpha)^n \sum_{i=K_1+1}^{K_1+k_2} p_{vi|b}^{t0} + 0 \approx 0 \tag{4.24}$$

Sustituyendo (4.22)-(4.24) a (4.21) y reordenando términos, uno puede obtener:

$$\alpha > 1-(1-M_1)^{\frac{1}{n}} \tag{4.25}$$

donde:

n: número de frames requeridos para aprender la nueva apariencia de background.

La ecuación (4.25) implica que si se desea que el nuevo sistema aprenda el nuevo estado de background en no más tarde que n frames, se escogerá α

de tal manera que (4.25) es satisfecha. Por ejemplo, si el sistema está para responder a un cambio repentino de background en 20 s con el ratio de frame siendo 20 fps y $M_1 = 85\,\%$,$\alpha > 0{,}00473$ sería satisfecho.

III. Detección de objeto de foreground: El algoritmo

Con la formulación bayesiana de la clasificación de background y foreground, además de la representación de background con características principales, se desarrolla un algoritmo para la detección de objetos de foreground en entornos complejos. Éste consta de cuatro pasos: detección de cambio, clasificación de cambio, segmentación de objeto de foreground, y mantenimiento de background. El diagrama de bloques del algoritmo se muestra en la figura 4.6.

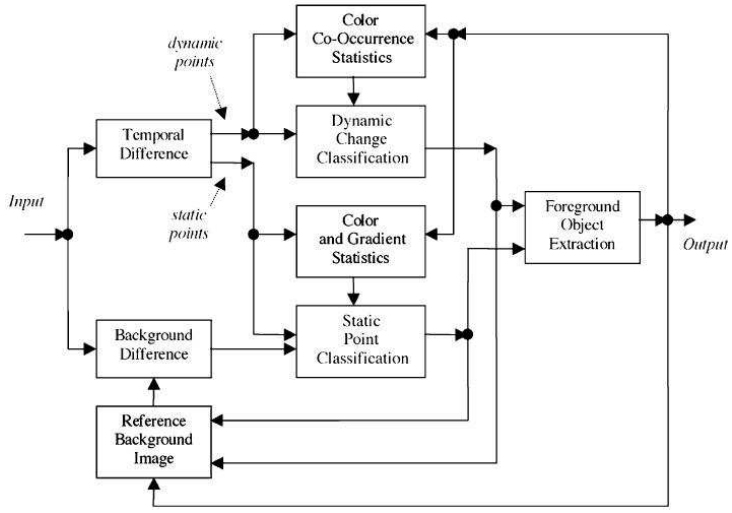

Figura 4.6: Diagrama del algoritmo FGD.

En primer lugar, los píxeles de background que se mantienen inalterados en el frame actual son filtrados hacia fuera mediante el uso de background simple y diferenciación temporal. Los cambios detectados son separados en puntos estáticos y dinámicos acorde a cambios de interframe. En el segundo paso, los puntos de cambio estáticos y dinámicos detectados son también clasificados como background o foreground usando la regla de Bayes y las estadísticas de las características principales para background. Los puntos estáticos son clasificados basándose en las estadísticas de los colores y gradientes principales, mientras que los puntos dinámicos son clasificados basándose en las estadísticas de co-ocurrencias de color principales. En el

tercer paso, los objetos de foreground son segmentados mediante la combinación de los resultados de clasificación de tanto los puntos estáticos como los puntos dinámicos. En el cuarto paso, los modelos de background son actualizados. Esto incluye actualizar las estadísticas de las características principales para el background además de una imagen de background de referencia. A continuación se describen los pasos brevemente:

A. Detección de cambio

En este paso, se usa la diferenciación de imagen adaptativa simple para filtrar hacia fuera píxeles de background inalterados. Las menores variaciones de color causadas por la ruido en la imagen son filtradas hacia fuera para guardar el cálculo para su posterior procesamiento.

Sea $I(s,t) = I_c(s,t)$ la imagen de entrada y $B(s,t) = B_c(s,t)$ la imagen de background referenciada mantenida en tiempo t con C$\in$ R,G,B denotando un componente de color. La diferencia de background se obtiene como sigue. Primero se realizan la diferenciación y el umbral de la imagen para cada componente de color, donde el umbral es automáticamente generado usando el método de menor mediada de cuadros (LMedS) método [20]. La diferencia de background $F_{bd}(s,f)$ es entonces obtenida mediante fundición de los resultados de los tres componentes de color. De manera similar, la diferencia temporal (o interframe) $F_{td}(s,t)$ entre dos frames consecutivos I(s,t-1) y I(s,t) es obtenida. Si $F_{bd}(s,t) = 0$ y $F_{td}(s,t) = 0$, el píxel es clasificado como un punto de background inalterado. En general, más del 50 % de los píxeles serían filtrados hacia fuera en este paso.

B. Clasificación de cambio

Si $F_{td}(s,t) = 1$ es detectado en un píxel s, éste es clasificado como un punto dinámico, de otra manera, es clasificado como un punto estático. Un cambio que ocurre en un punto estático podría ser causado por cambios de iluminación, cambios de background repentinos, o un objeto de foreground inmóvil temporalmente. Un cambio detectado en un punto dinámico podría ser causado por un objeto de background o foreground moviéndose. Estos son también clasificados como background o foreground mediante el uso de la regla de decisión de Bayes y las estadísticas de las características principales correspondientes.

Sea v_t el vector característico de entrada en s y tiempo t. Las probabilidades son estimadas como:

$$\begin{aligned} P_s(b) &= p_v^t(b) \\ P_s(v_t) &= \sum vj \in U(vt) P_{vj}^t \\ P_s(v_t \mid b) &= \sum vj \in U(vt) P_{vj|b}^t \end{aligned} \tag{4.26}$$

donde:

$U(v_t)$ es una serie de vectores característicos compuestos en $T_v(s)$ que casan con el vector de entrada V_t, por ejemplo:

$$U(vt) = \{v_j \in T_v(s), d(v_t, v_j) \leq \sigma \text{ y } j \leq N(v)\} \tag{4.27}$$

Si no hay un vector característico principal en la tabla $T_v(s)$ que empareje con V_t, ambos $P_s(v_t)$ y $P_s(vt|b)$ son puestos a 0. Entonces, el punto de cambio es clasificado como background o foreground como sigue.

Clasificación de punto estático

Para un punto estático, las probabilidades para tanto las características de gradiente como de color son estimadas mediante (4.26) con v=c y v=e, respectivamente, donde la medida de distancia de vector $d(v_1, v_2)$ en (4.27) es calculada como (4.10). En este trabajo, los estadísticos de las dos características principales ($T_c(s) y T_e(s)$) son aprendidos por separado. En casos generales, sería $p_c^t(b) \approx p_e^t(b)$. La regla de decisión de Bayes (4.9) puede ser aplicada para la clasificación de background y foreground.

En algunos casos complejos, un tipo de las características del fondo puede ser inestable. Un ejemplo son los estados temporales estáticos de una superficie agua oscilante. Para estos estados, las características de gradiente no son constantes. Otro ejemplo es el vídeo capturado con una cámara automática de ganancia. La ganancia es a menudo autocompensada debido al movimiento de objetos y las características de gradiente son más estables que las características de colores para los píxeles estáticos del fondo. Sea $P_s(b)$ = max($p_c^t(b)$,$p_e^t(b)$) y $P_m(b)$=min($p_c^t(b)$,$p_e^t(b)$). Si $P_m(b) > KP_s(b)$(K=0.6 en la prueba), las características de color coinciden y ambas características se utilizan para la clasificación utilizando la regla de Bayes 4.9. De lo contrario, sólo un tipo de las características con un valor mayor $p_v^t(b)$ se utiliza para la clasificación utilizando la regla de Bayes 4.5.

Clasificación de punto dinámico

Para un punto dinámico en el tiempo t, el vector característico de co-ocurrencia de color cc_t es generado. Las probabilidades para cc_t son calculadas como en (4.26), donde la distancia entre dos vectores característicos en (4.27) es calculada como:

$$d(cc_t, cc_j) = max_{k\in[1,6]}\{|cc_{tk} - cc_{jk}|\} \tag{4.28}$$

y $\sigma = 2$ es elegido. Finalmente, la regla de Bayes (4.5) es aplicada para clasificación de background y foreground. Generalmente, para los puntos de background dinámicos, sólo un pequeño porcentaje de ellos son clasificados erróneamente como cambios de foreground. Además, los restantes han llegado a ser puntos aislados, que pueden ser fácilmente eliminados por medio de una operación suave.

C. Segmentación de objeto de foreground

Se aplica un proceso posterior para segmentar los puntos de cambio restantes hacia regiones de foreground. Ésto es hecho primeramente mediante la aplicación de una operación morfológica (una pareja de abrir y cerrar) para suprimir los errores residuales. Luego las regiones de foreground son extraídas, los agujeros son rellenados y las regiones pequeñas son eliminadas. Además una operación AND es aplicada a los segmentos resultantes en frames consecutivos para eliminar las regiones de foreground falsas detectadas mediante diferenciación temporal.

D. Mantenimiento de background

Con la realimentación de la segmentación anterior, los modelos de background son actualizados. Primero, las estadísticas de las características principales son actualizadas como se describió anteriormente. Para los puntos estáticos, las tablas $T_c(s) y T_e(s)$ son actualizadas. Para los puntos dinámicos, la tabla $T_{cc}(s)$ es actualizada. Mientras tanto, una imagen de background de referencia es también mantenida para hacer la conveniente diferencia de background. Sea s un punto de background en el resultado de segmentación final en el tiempo t. Si es identificado como un punto de background inalterado en el paso de detección de cambio, la imagen de referencia de background en s es suavemente actualizada por:

$$B_c(s, t+1) = (1-\beta)B_c(s,t) + \beta l_c(s,t) \tag{4.29}$$

donde:

C $\in R, G, B$

β: número positivo pequeño.

Si s es clasificado como background en el paso de clasificación de cambio, la imagen de referencia de background en s es reemplazada por la nueva apariencia de background:

$$B_c(s, t+1) = l_c(s,t) \text{ para C} \in \{R, G, B\} \tag{4.30}$$

Con (4.30), la imagen de background de referencia puede seguir los cambios de background dinámicos, por ejemplo, los cambios de color entre la rama de árbol y el cielo, además de cambios de background repentinos.

Algoritmo MOG

En este método o algoritmo se modelan los valores de un píxel particular como una mezcla de distribuciones gaussianas. Basándose en la persistencia y la concordancia de cada una de las gaussianas de la mezcla, se determina cuáles de ellas podrían corresponder a colores de background. Los valores de píxel que no encajan con distribuciones de background son considerados foreground hasta que haya una gaussiana que los incluya con evidencia suficiente.

En primer lugar se debe saber que si un píxel resulta de una superficie concreta bajo una iluminación determinada, una distribución gaussiana simple sería suficiente para modelar el valor del píxel. Por otro lado, si solamente se producen cambios de iluminación durante horas, una distribución gaussiana simple y adaptativa sería suficiente. Pero en la práctica, son múltiples superficies las que aparecen a menudo en la vista de un píxel particular y las condiciones de iluminación cambian. Por lo tanto, en ese caso, son necesarias gaussianas múltiples adaptativas. Este método usa entonces una mezcla de gaussianas adaptativas para aproximar este proceso.

Cada cierto tiempo los parámetros de las gaussianas son actualizados. Las gaussianas son evaluadas usando una simple heurística para hacer hipótesis de cuáles son más probables que sean parte del proceso de background. Los valores del píxel que no concuerdan con los píxeles del background son agrupados usando componentes relacionados. Finalmente, los componentes relacionados son rastreados frame a frame usando un rastreador de hipótesis múltiple. El proceso es relatado a continuación:

1) Modelo Mixto on-line

Se considera a los valores de un píxel particular durante un período de tiempo como un «proceso de píxel». El «proceso de píxel» es una serie de tiempo de valores del píxel. Por ejemplo escalares para valores de gris o vectores para imágenes en color. En cualquier momento, t, lo que se sabe acerca de un píxel en particular, (x0, y0), es su historia:

$$\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 < i < t\} \tag{4.31}$$

donde:

I es la secuencia de imágenes.

Algunos procesos de «píxeles» son mostrados por diagramas de dispersión en la figura 4.7, los cuales ilustran la necesidad de un sistema adaptativo con una umbralización automática.

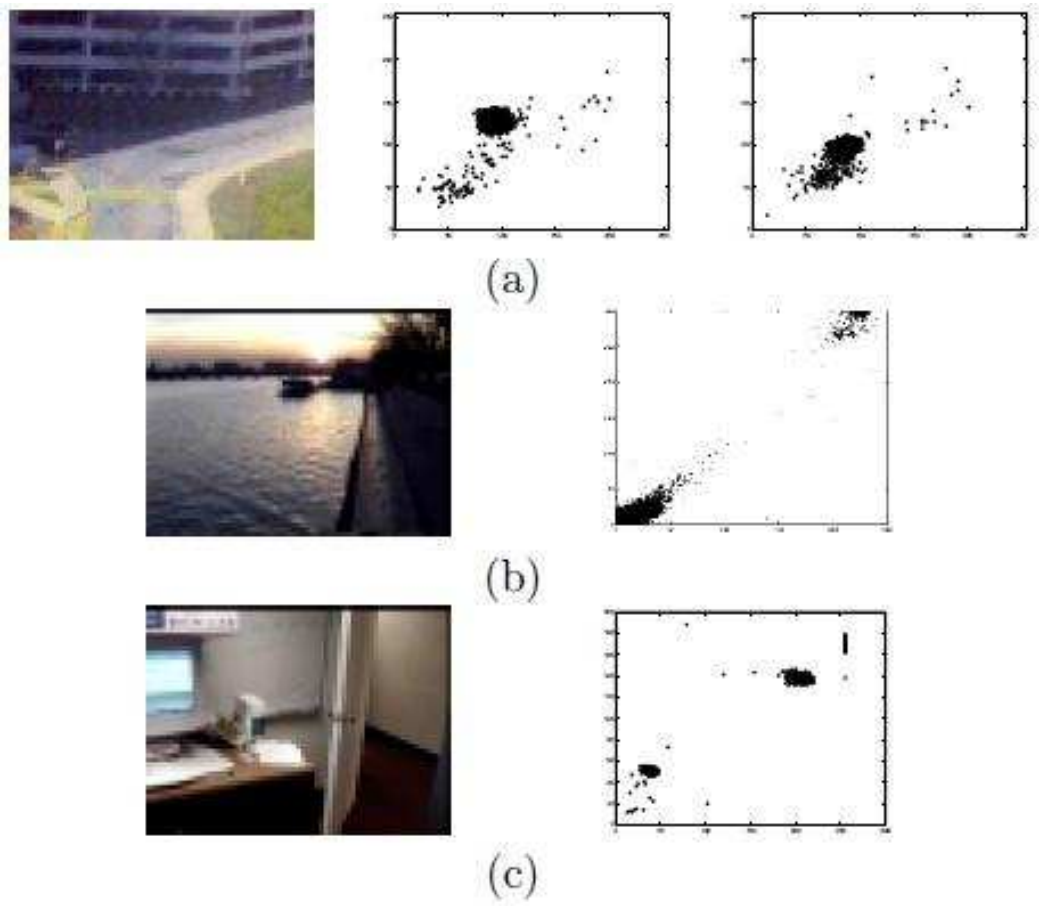

Figura 4.7: Esta figura contiene imágenes y diagramas de dispersión de los valores de rojo y verde de un píxel de la imagen.

El valor de cada píxel representa una medida de la claridad en la dirección del sensor del primer objeto intersectado por los rayos ópticos del píxel. Con un fondo e iluminación estáticos, dicho valor es relativamente constante. Si suponemos la independencia del ruido gaussiano que incurre en el proceso de toma de muestras, su densidad puede ser descrita por una sola distribución de Gauss centrada en el valor del píxel. Por desgracia, las secuencias de vídeo más interesantes involucran cambios de luz, cambios de escenas, y objetos moviéndose.

Si se producen cambios de iluminación en una escena estática, sería necesario para la gaussiana rastrear esos cambios. También puede darse el caso de que un objeto estático sea añadido a la escena y no sea incorporado al

background hasta que esté allí más tiempo que el objeto previo, por lo que los píxeles correspondientes serían considerados foreground durante largos períodos de tiempo. Esto conduciría a acumular errores en la estimación del foreground, resultando pobres comportamientos de rastreo. Estos factores proponen que las observaciones más recientes deberían ser más importantes a la hora de determinar las estimaciones de parámetros gaussianos.

Un aspecto adicional de variación ocurre si en las escena están presentes objetos moviéndose, ya que un objeto en movimiento coloreado relativamente de forma constante se espera que produzca más variación que un objeto «estático». Además, en general, habrá más datos dando soporte a las distribuciones de background porque éstas son repetidas, mientras que los valores de píxel para diferentes objetos no son a menudo del mismo color.

Todos estos factores mencionados son los que sirven de guía a la hora de de modelar y actualizar el procedimiento. La historia reciente de cada píxel, {X1,...,Xt}, está modelada por una mezcla de K distribuciones gaussianas. La probabilidad de observar el valor del píxel actual es:

$$P(X_t) = \sum_{i=1}^{K} W_{i,t} * \eta(X_t, \mu_{i,t}, \sum i, t) \tag{4.32}$$

donde:

K, es el número de distribuciones.

$W_{i,t}$ es una estimación del peso de la i^{th} gaussiana en el momento t.

$\mu_{i,t}$ es el valor de la i^{th} gaussiana en el momento t.

$\sum i, t$ es la matriz de covarianza de la i^{th} gaussiana en el momento t.

y donde η es una gaussiana dada por la función de densidad:

$$\eta(X_t, \mu, \sum) = \frac{1}{(2\pi)^{\frac{n}{2}} |\sum|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \sum^{-1} (X_t - \mu_t)} \tag{4.33}$$

K es determinado por la memoria disponible y la capacidad de cómputo. En la actualidad, se utilizan de 3 a 5 . Por otra parte, por razones de cálculo,

la matriz de covarianza se suponen de la forma:

$$\sum k, t = \sigma_k^2 I \tag{4.34}$$

Esto supone que los valores de los píxeles de color rojo, verde y azul son independientes y tienen las mismas variaciones. Esto no es exactamente cierto, pero permite evitar una costosa inversión de la matriz a expensas de cierta exactitud.

Así pues, la distribución de los valores observados recientemente de cada píxel en la escena está caracterizada por una mezcla de gaussianas. Un nuevo valor de píxel estará, en general, representado por uno de los componentes principales del modelo mixto y será usado para actualizar el modelo.

Si el proceso de píxel fuera considerado un proceso estacionario, un método estándar para maximizar la probabilidad del dato observado es el expectation maximation. Desafortunadamente, como ya se dijo anteriormente, cada proceso de píxel varía durante el tiempo como el estado del mundo cambia, por lo que se usará un método aproximado que trata esencialmente cada nueva observación como una colección de muestra de tamaño 1 y usa reglas de aprendizaje estándar para integrar la nueva información.

Debido a que hay un modelo mixto para cada píxel en la imagen, implementar un algoritmo EM exacto sería costoso. En su lugar, se implementa una aproximación on-line de k medias. Cada nuevo valor del píxel, Xt, es cotejado con las distribuciones K gaussianas existentes, hasta que se encuentra coincidencia. Una coincidencia es definida como un valor de píxel dentro de una desviación estándar de 2.5 de una distribución. Este umbral puede ser perturbado con un pequeño efecto en la ejecución. Esto es realmente un umbral por píxel por distribución, y es extremadamente útil cuando diferentes regiones tienen diferente iluminación, porque los objetos que aparecen en regiones sombreadas no exhiben generalmente tanto ruido como los objetos en las regiones iluminadas. Un umbral uniforme a menudo resulta en objetos que desaparecen cuando éstos entran en regiones oscuras.

Si ninguna de las K distribuciones empareja con el valor del píxel actual, la distribución menos probable es reemplazada por una distribución con el valor actual como su valor principal, una varianza alta inicialmente, y un peso de prioridad bajo.

Los pesos de prioridad de las K distribuciones en el instante t son ajustados como sigue.

$$W_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t}) \tag{4.35}$$

donde:

α es la tasa de aprendizaje.

$M_{k,t}$ es 1 para los casos que casan y 0 para el resto de modelos.

Después de esta aproximación, los pesos son renormalizados.$1/\alpha$ define la constante de tiempo que determina la velocidad a la cual cambian los parámetros de las distribuciones. $W_{k,t}$es en realidad un filtro paso bajo de los valores promedio de la probabilidad posterior de que los valores de píxel hayan encajado en el modelo k en observaciones de 1 a t. Esto es equivalente a la expectativa de este valor con una exponencial basada en los valores del pasado.

Los parámetros μ y σ para distribuciones que no casan se quedan igual. Los parámetros de la distribución que casan con la nueva observación son actualizados como sigue.

$$\mu_t = (1 - \rho)\mu_{t-1} + \mu X_t \tag{4.36}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{4.37}$$

$$\rho = \alpha\eta(X_t \mid \mu_k, \sigma_k) \tag{4.38}$$

Una de las ventajas importantes de este método es que si algo pasa a formar parte del background, esto no destruye el modelo existente de background, ya que el color de background original permanece en la mezcla hasta que llega a ser el K^{th} más probable y un nuevo color es observado. Por lo tanto, si un objeto es estacionario lo suficientemente largo para formar parte del background y entonces se mueve, la distribución describe el background

previo hasta que exista con el mismo μ y σ^2, pero un bajo w y será rápidamente reincorporada a el background.

2) Estimación del modelo de background

Como los parámetros del modelo mixto de cada píxel cambian, se quiere determinar cuál de las gaussianas de la mezcla son producidas más probablemente por los procesos de background. Interesa la distribución gaussiana que tenga la evidencia de mayor soporte y menor variación.

Para comprender esta elección, se considera la acumulación de evidencias de soporte y la variación relativamente baja para las distribuciones de «background» cuando un objeto estático y persistente es visible. En contraste, cuando un objeto nuevo obstruye el objeto de background, este, en general, no casará con ninguna de las distribuciones existentes e implicará o bien la creación de una nueva distribución o en el incremento de la variación de una distribución existente. Además, la variación de un objeto en movimiento se espera que permanezca más larga que un píxel de background hasta que el objeto en movimiento pare. Para modelar esto, se necesita un método para decidir que porción del modelo mixto es la que mejor representa los procesos de background.

Primero, las gaussianas son ordenadas por el valor w/σ. Este valor aumenta tanto por la distribución de ganancias como por disminuir la varianza. Después de re-estimar los parámetros de la mezcla, tenemos lo suficiente para emparejar la distribución con el fondo de distribución más probable. Esta ordenación del modelo es efectivamente una lista ordenada y sin límite, donde las distribuciones de background más probables permanecen en la cima y las distribuciones de background pasajeras menos probables gravitan hacia el fondo y son eventualmente sustituidas por nuevas distribuciones.

Entonces las primeras B distribuciones son elegidas como el modelo de background, donde:

$$B = argmin_b(\sum_{k=1}^{b} W_k > T) \tag{4.39}$$

donde:

T es una medida de la porción mínima de la información que debería ser explicada por el background. Esto toma la «mejor» distribución hasta una porción cierta, T, de la reciente información que ha sido explicada. Si se escoge un pequeño valor para T, el modelo de background es normalmente unimodal. Si este es el caso, usando solamente la distribución más probable salvaremos el proceso.

Si T es más alto, una distribución multimodal causada por un movimiento de background repetitivo (por ejemplo, las hojas de un árbol, una bandera en el viento, etc.) resultaría en que más de un color sería incluido en el modelo de background. Estos resultados permiten al background aceptar dos o más colores separados.

3) Componentes conectados

El método descrito anteriormente permite identificar píxeles de foreground en cada nuevo frame mientras se actualiza la descripción de cada proceso de píxel. Estos píxeles de foreground etiquetados pueden entonces ser segmentados en regiones por un algoritmo de componentes conectados en dos pasos.

Ya que este procedimiento es efectivo en determinados objetos en movimiento, las regiones en movimiento pueden ser caracterizadas no sólo por su posición, si no también por su tamaño, momentos (instantes, importancias), y otra información de forma. No sólo pueden estas características ser útiles para el posterior proceso y clasificación, si no que también pueden ayudar en el proceso de rastreo.

4) Rastreo de hipótesis múltiple

Estableciendo correspondencia de los componentes conectados entre frames se logra usar un algoritmo de rastreo de múltiples hipótesis predictivo directamente que incorpore tanto posición como tamaño. Debido a que este algoritmo no es esencial en el entendimiento del método, es preferible no profundizar en el funcionamiento del mismo debido a su complejidad.

4.2.2. Parámetros de operación de los algoritmos

Los algoritmos se diseñan de tal manera que una serie de parámetros se pueden cambiar en busca de la combinación más óptima, de manera que se

pueden adaptar de la mejor manera posible a cada posible entorno y condiciones de trabajo. Como se ve en la explicación teórica, ambos algoritmos dependen de algunos valores, añadidos a otros que se introducen, como mínima área necesaria para interpretar que se trata de un objeto de foreground, conforman un grupo de datos que se pueden modificar para así optimizar el resultado obtenido. A continuación se muestran todos los parámetros de los que depende cada algoritmo.

Algoritmo FGD

- Lc: Niveles cuantizados para el componente de color. Potencia de dos, normalmente 32, 64 o 128.
- N1c: Número de vectores de color usados para modelar la variación de color del background normal en un píxel dado.
- N2c: Número de vectores de color mantenidos en un píxel dado. Tiene que ser mayor que N1c, normalmente alrededor de 5/3 de N1c. Usado para permitir a los primeros N1c vectores adaptarse a cambios de background que ocurran con el paso del tiempo.
- Lcc: Niveles cuantizados para el componente de co-ocurrencia de color. Potencia de dos, normalmente 16, 32 o 64.
- N1cc: Número de vectores de co-ocurrencia de color usados para modelar la variación de color del background normal en un píxel dado.
- N2cc: Número de vectores de co-ocurrencia de color mantenidos en un píxel dado. Tiene que ser mayor que N1cc, normalmente alrededor de 5/3 de N1cc. Usado para permitir a los primeros N1cc vectores adaptarse a cambios de background que ocurran con el paso del tiempo.
- is_obj_without_holes: Si tiene valor TRUE se ignoran agujeros dentro de los blobs de foreground. Valor típico: TRUE.
- perform_morphing: Número de iteraciones de limpieza sobre los blobs de foreground que se deforman continuamente. Normalmente se pone 1.
- alpha1: Indica cómo de rápido se olvidan los valores de un píxel viejo de foreground. Normalmente este parámetro es puesto a 0.1.
- alpha2: Controles de velocidad de la función de aprendizaje. Depende del parámetro T. Su valor normalmente ronda el 0.005.

- alpha3: Suplente de alpha2, usado (por ejemplo) para una rápida convergencia inicial. Su valor típico es el de 0.1.
- delta: Afecta a la cuantización de color y co-ocurrencia de color: Este parámetro normalmente es puesto a 2.
- T: Un valor porcentual que determina cuando nuevas características pueden ser reconocidas como nuevo background. Normalmente 0.9.
- minArea: Descarta blobs de foreground cuya caja (área donde queda recogido un objeto de foreground) es más pequeña que este umbral.

Algoritmo MOG

- win_size: Determina el valor de la constante de aprendizaje (alpha), ya que alpha=1/win_size.
- n_gauss: Número de distribuciones gaussianas en la mezcla. Normalmente, su valor oscila entre 3 y 5.
- bg_threshold: Umbral suma de los pesos para los test de background.
- std_threshold: Umbral de desviación estándar (lambda). Normalmente su valor es 2.5.
- minArea: Descarta blobs de foreground cuya caja (área donde queda recogida un objeto de foreground) es más pequeña que este umbral.
- weight_init: Peso inicial de la distribución gaussiana.
- variante_init: Desviación estándar inicial de la distribución gaussiana.

4.2.3. Elección del algoritmo para la detección de fondo

Una vez que se estudia detalladamente los dos algoritmos principales es necesario elegir uno de los dos para utilizar en la aplicación. Evidentemente la mejor manera de decidir cual es el mejor es mediante la experimentación, es decir, probando ambos algoritmos en todos los lugares posibles. Además

hay que tener en cuenta que cada algoritmo depende, a su vez, de los parámetros de entrada que se acaban de mencionar en el apartado 4.2.2. Por tanto, hay muchísimas alternativas a estudiar para cada algoritmo en cada entorno. Únicamente hacer una experimentación exhaustiva del funcionamiento de ambos algoritmos en los posibles entornos de trabajo para elegir el más óptimo es materia más que suficiente para realizar otro proyecto. Debido a esto aqui se utiliza la bibliografía existente sobre análisis del comportamiento de estos algoritmos en diferentes lugares. En [16] se realiza un estudio detallado sobre el comportamiento de ambos algoritmos en distintas situaciones. A continuación se muestran las principales conclusiones.

- Incorporación de background a la escena (objeto que pasado un tiempo se incorpora al fondo): Ambos algoritmos tienen un comportamiento óptimo. Con los parámetros preestablecidos en el MOG el tiempo es menor, pero en FGD este varía cambiando el parámetro alpha 2, pudiendo llegar a ser menor.

- Background dinámico (por ejemplo movimiento de hojas de un árbol): Correcto funcionamiento por lo general en FGD que no detecta como foreground este movimiento. Por el contrario en MOG en más ocasiones si se detecta este background dinámico como foreground.

- Cambios de iluminación: En el FGD no se detectan las zonas afectadas por el cambio de iluminación como foreground, mientras que en el MOG en ocasiones si.

- Vídeo comienza con foreground en escena: Ambos tienen comportamiento similar.

- Foreground estático (hace referencia a los blobs que por una razón u otra se encuentran parados en un momento dado): Ambos algoritmos presentan un comportamiento similar salvo el caso en el que una persona esta parada y realiza pequeños movimientos (por ejemplo una dependienta atendiendo a un cliente en una tienda), en este caso el comportamiento del MOG es mejor.

Figura 4.8: Respuesta de los algoritmos ante foreground estático con pequeños movimientos.

- Foreground dinámico y solitario:
    - Sin paradas: En este caso a distancias cortas y medias ambos algoritmos detectan el movimiento, si bien el algoritmo FGD representa la silueta de manera mucho más real, es decir, con mucho menos ruido. Por el contrario, a largas distancias si la cámara no tiene una elevada resolución el algoritmo MOG presenta mejores prestaciones a la hora de detectar el movimiento.

Figura 4.9: Respuesta de los algoritmos ante foreground a largas distancias. Se comprueba como la respuesta del algoritmo MOG es mejor que la respuesta del algoritmo FGD.

    - Con paradas: Comportamiento similar en ambos casos salvo cuando el color del foreground es muy parecido al color del background en cuyo caso el MOG es algo mejor.
- Foreground dinámico y en grupo: Resultados similares en ambos casos.
- Personas que se cruzan: Comportamiento parecido también en ambos casos.

Una vez vistas las conclusiones obtenidas sobre las características de ambos algoritmos en ciertas circunstancias y en determinados vídeos (puede que en otros vídeos los resultados obtenidos fueron otros), se debe decidir que algoritmo utilizar. Dado que en nuestra aplicación se busca obtener buenos resultados, sobretodo a medias distancias, se decide utilizar el algoritmo FGD. Este algoritmo detecta los contornos de una manera mucho más fiel a la realidad pues tiene bastante menos ruido, lo cual es muy importante para no hacer detecciones falsas, bien debidas a cambios de iluminación, movimientos de árboles o simplemente porque no se trata de una persona. Por el contrario su principal problema es que puede que se deje de representar personas en el foreground que se encuentran muy lejanas, pero este no es un caso importante en nuestra aplicación que está destinada principalmente para la interacción robot-humano. También puede que deje de representar personas en el foreground cuyo movimiento, una vez que está estática, es mucho menor o su color es muy similar al del background. A continuación se muestra el foreground detectado para el mismo frame de un vídeo para poder observar que no compensa los pequeños casos extra que permite detectar MOG en proporción al ruido que introduce muchas veces a medias distancias.

Figura 4.10: Respuesta de los algoritmos ante movimiento a medias distancias. Se puede observar como en el algoritmo MOG hay más ruido debido a las sombras, esto va a provocar que se detecte, o bien dos personas una encima de otra, o una sola persona del doble de altura, esto evidentemente va a engañar al robot.

Después de decidir que el algoritmo para la extracción de fondo a utilizar será el FGD, es necesario determinar cuales son los valores más adecuados para sus parámetros. Los valores estándar para este algoritmo son:

- Lc=64
- Lcc=32

- alpha1=0.1
- alpha2=0.005
- alpha3=0.1
- N1c=30
- N2c=50
- N1cc=50
- N2cc=80
- is_obj_without_holes=TRUE
- perform %morphing=1
- delta=2
- T=0.9
- minArea=15

En [16] se dice, que bajo la experimentación concreta realizada sobre una serie de vídeos, los valores mejores para entornos exteriores que cambian respecto a los estándar son:

- Lc=128
- Lcc=64
- alpha2=0.1

Mientras que para entornos interiores se aconsejan introducir los valores estándar.

Cabe destacar que esto es con el único objetivo de representar el movimiento de la manera más fidedigna posible, pero para nuestra aplicación lo que nos interesa es representar a las personas que hay con la mayor exactitud posible. Por lo tanto puede que otros valores sean más aconsejables, pues puede que por ejemplo introduzcan un poco más de ruido que después se puede eliminar pero la persona permanezca más tiempo detectada.

En conclusión, cuando se quiera probar esta aplicación en un determinado lugar se aconseja introducir estos valores dependiendo de si es interior o exterior para después hacer un ajuste más fino en el entorno en concreto para conseguir unos resultados optimizados. Este es el método que se sigue para realizar la experimentación con el algoritmo en el capítulo 5.

4.2.4. Filtro Morfológico

Las transformaciones morfológicas modifican la forma de los objetos de una imagen. En este proyecto son de gran ayuda, pues en la imagen de foreground resultante del módulo anterior hay partes que se ven afectadas por la generación de regiones o píxeles con ruido. Es decir, se obtienen zonas de movimiento falsas donde realmente no hay movimiento, o el movimiento es mínimo, de manera que se le considera despreciable. Para solucionar este problema se recurre a las operaciones de filtrado morfológico, estás son las de erosión y dilatación.

Erosión

Es la degradación progresiva de uno de los campos (0 ó 1). Un elemento del campo a degradar seguirá perteneciendo al mismo si está rodeado de elementos iguales a él, en caso contario pasará al otro campo. Matemáticamente se expresará de la siguiente forma: si se toma el elemento estructural simétrico respecto al origen de B, $\breve{B}$, la erosión de un conjunto X respecto al elemento B es:

$$X \ominus \breve{B} = \{x \mid B_x \subset X\} \tag{4.40}$$

Lo que es igual a una transformación acierta o falla donde B_x^2 es el conjunto vacío. El símbolo $\ominus$ representa la resta de Minkowski.

Dilatación

Es el crecimiento progresivo de uno de los campos (0 ó 1). Un elemento del campo contrario a crecer será convertido si posee algún vecino perteneciente al campo que se expansiona. En caso contrario, permanecerá igual. Los elementos pertenecientes al campo a expansionar evidentemente no se modifican. Si se aplicase un número elevado de veces terminaría por destruir la imagen ya que todos lo píxeles estarían a nivel alto. Matemáticamente se puede expresar la dilatación como:

$$X \oplus \breve{B} = (X^c \ominus \breve{B})^c \tag{4.41}$$

Combinación de ambas

Una de las principales características de la dilatación y la erosión es que no cumplen la propiedad conmutativa, es decir, no es lo mismo realizar una dilatación seguida de una erosión, que una erosión seguida de una dilatación. En este proyecto se ejecuta en primer lugar dos repeticiones para la erosión y a continuación una para la dilatación. Cuando las operaciones se realizan en este orden el proceso recibe el nombre de apertura. Se denomina así ya que al empezar por una erosión se tiende a romper las piezas en sus partes constitutivas. Las principales causas por las que se elige este orden y no el inverso es que se consigue suavizar los contornos del objeto, se rompen enlaces delgados y se eliminan pequeñas protuberancias. A continuación se muestra el resultado de aplicar el filtro morfológico.

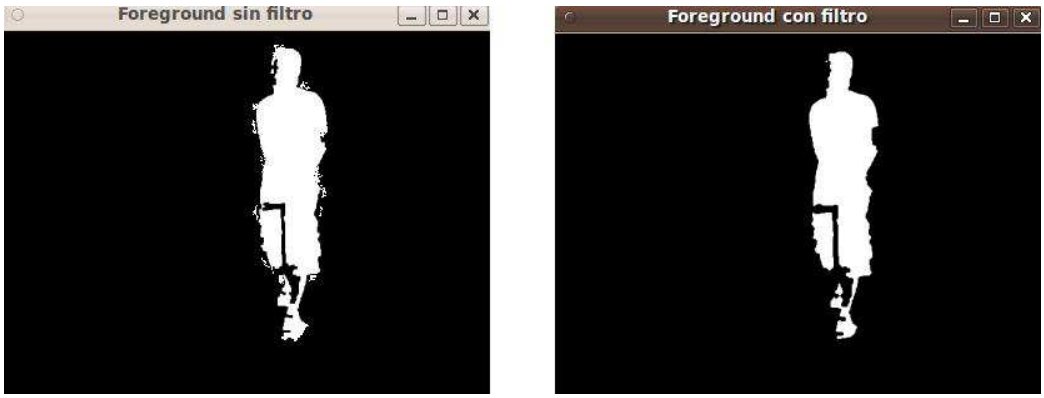

Figura 4.11: Aplicación del filtro morfológico.

4.3. Detección de blobs

A continuación se analiza como se realiza la detección de blobs en movimiento en general, para en el siguiente apartado determinar como se selecciona solo los que representan personas.

En este módulo se utiliza como entrada la salida del módulo anterior (máscara de FG). En cada frame se agrupan los píxeles en movimiento que

se encuentran continuos o adyacentes entre sí como una sola región (blob). Dado que la máscara de foreground es una imagen binaria no es necesario realizar una umbralización (convierte una imagen con varios niveles de grises a una nueva con sólo dos). En la imagen que se trabaja los píxeles blancos son aquellos que representan las zonas en movimiento y los negros las zonas inmóviles. Para realizar la agrupación de los blobs (detección) por separado se utiliza la técnica de etiquetado.

Figura 4.12: Detección de una persona.

El etiquetado (del inglés labelling) consiste en, empezando por el píxel superior izquierdo (figura 4.13), ir recorriendo la imagen en dirección a la derecha. Cuando se encuentra el primer píxel que esté a nivel alto se le asocia la etiqueta (del inglés label) 1, para identificarlo como primer objeto. Se examinan a continuación sus vecinos para ver si también están a nivel alto, si lo están reciben la misma etiqueta. Cuando un píxel no sea vecino de uno etiquetado pero esté a nivel alto se le asocia la siguiente etiqueta, 2 y así sucesivamente. Con un solo recorrido de la imagen se pueden producir indeterminaciones. La solución está en llevar una tabla con etiquetas que corresponden al mismo objeto. Una vez terminada esta segunda pasada sobre la imagen se deshacen las ambigüedades asignando a todos los puntos la misma etiqueta.

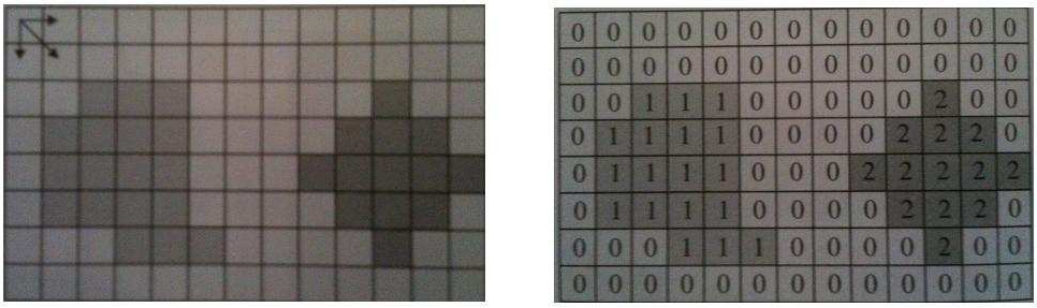

Figura 4.13: Algoritmo de etiquetado.

4.4. Detección de personas

En este bloque lo que se pretende es que una vez se obtiene una serie de blobs determinar si estos pertenecen a personas o no, ya que este es el principal objetivo de la aplicación. Para conseguir realizar está segmentación existen distintas técnicas como se puede ver en el capítulo 2.

La primera técnica que se ve en ese capítulo está basada en el análisis de contornos. Este método no es conveniente, porque si bien es el más preciso para detectar las posturas introducidas en la biblioteca de contornos, presenta muchos problemas para el resto. Debido a que la posturas de las personas a detectar por esta aplicación no son siempre la misma, es decir, no se pretende que solo se detecten personas erguidas, de frente, sin ruido y de silueta completa. Con esto se quiere decir que si por ejemplo una persona se mueve por detrás de un objeto de background estático, con lo que solo se le ve de cintura para arriba, también se le quiere detectar, o si aparecen dos personas unidas en la escena, también se les quiere detectar, etc. Por lo tanto la biblioteca de posturas tendría que ser muy extensa y el coste computacional muy elevado.

La segunda de las técnicas consiste en un análisis de regiones calculando de modo iterativo la elipse más grande contenida en cada región estática obtenida después de la etapa de segmentación, este tipo de algoritmo tiene un gran porvenir pero hoy en día falla en demasiados casos por motivos similares a los explicados para la primera técnica.

La tercera de las técnicas es la seleccionada para la detección de personas

en está aplicación, este método esta basado en la utilización de las características de los blobs. En un primer lugar se hace un filtrado de áreas, de tal manera que las áreas que este por debajo de un valor o por encima de otro se filtran. Se supone que se pretende detectar personas en un determinado rango de distancia, por tanto áreas menores o mayores que los límites de detección se filtran (por ejemplo vehículos o pequeñas detecciones por cambios de iluminación). El siguiente filtrado se realiza de acuerdo a la relación entre el largo y el ancho del blob detectado, el cual para personas erguidas se ha demostrado empíricamente sigue una gaussiana de media 0.3 y varianza 0.2, dado que también se quiere detectar personas sentadas en el algoritmo o dos personas que aparecen de la mano en escena por ejemplo, se amplía el valor límite superior. Gracias a esta restricción se dejan de detectar objetos cuyo ancho sea elevado en proporción a su alto. Por último se tiene en cuenta que la proporción del área del blob detectado respecto a la caja englobante que lo envuelve se ha demostrado empíricamente también que para personas es entorno al 70 % (dejando de detectar con esta restricción objetos semejantes o muy diferentes a la caja englobante que lo envuelve). Con todas estas restricciones se consigue una buena proporción de detecciones correctas, si bien no es 100 % fiable. A continuación se muestran distintos ejemplos donde el algoritmo filtraría esos objetos en movimiento debido a no cumplir los requisitos expuestos.

Figura 4.14: Objetos en movimiento que son filtrados por la aplicación.

Es aconsejable, igual que se comenta en el apartado 4.2.3 para la elección de los parámetros del algoritmo de segmentación, utilizar estos datos mostrados como base para después hacer un ajuste más fino de acuerdo a las condiciones de trabajo en cada caso. Por ejemplo si se quieren detectar

personas agachadas hay que ampliar el rango de la relación ancho/alto para que se detecten. O si solo se quieren detectar personas muy cercanas se debe aumentar el límite inferior de áreas filtradas. En el apartado 5 se realiza un ensayo de pruebas para estudiar el algoritmo en el que se determina los mejores valores para cada vídeo en concreto.

Una vez que se ha detallado como se realiza la detección de personas en cada frame es aconsejable recuperar el concepto de seguimiento global introducido en el capítulo 1 para tener un completo entendimiento de lo que es, y no es capaz de realizar esta aplicación. Como se menciona en dicho capítulo este programa puede realizar el seguimiento de una sola persona, esto es, saber en cada momento la posición de la persona pues solo consiste en hacer la detección en cada frame de la misma.

Figura 4.15: Seguimiento de una persona.

Pero en cambio cuando hay más de una persona la aplicación es capaz de decir en este frame hay n personas en n posiciones y en el siguiente frame hay el mismo número de personas en estas nuevas posiciones. Pero es incapaz de decir la persona cuyo centroide anteriormente estaba en x=2500 e y=1000 ahora está en x=2520 e y=1300 por ejemplo.

4.5. Generación de vídeo de salida

El siguiente módulo es el encargado de la generación del vídeo de salida, que es el resultado de toda la aplicación. Este bloque se ejecuta durante todo el programa y con él se muestran los resultados en el momento de la ejecución.

El proceso de generación del vídeo de salida, a partir de la recopilación de imágenes, es un proceso sencillo que se realiza de forma automática. La aplicación relatada en este proyecto muestra por pantalla 3 ventanas, una principal y dos secundarias. En la ventana principal se muestra en cada instante la imagen de entrada, bien sea en tiempo real o de un vídeo grabado, con una caja englobante rodeando a la persona detectada en cada frame. Esto se hace con la finalidad de que la aplicación sea más visual para el usuario, bien para que el vigilante se percate visualmente de que hay una persona en la zona video-vigilida o bien para que la persona que lo utiliza para la retroalimentación de un robot vea fácilmente que la persona está siendo detectada. Pero en realidad la información importante: número de personas detectadas en cada instante, posición, tamaño,.. se almacena en la aplicación para que el usuario las pueda utilizar a su antojo bien sea para realizar una tarea con un robot o para activar automáticamente una alarma. Por último indicar que las dos ventanas secundarias que se muestran son, por un lado el background y por otro el foreground después del filtrado morfológico, de manera que el usuario pueda observar como está funcionando la aplicación y porque la detección está siendo correcta o no.

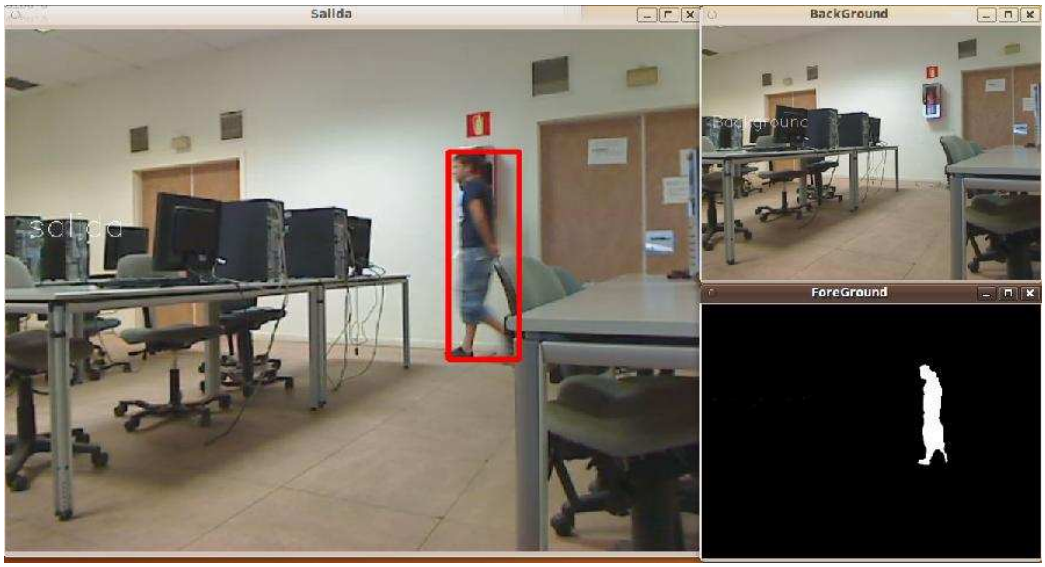

Figura 4.16: Pantalla de salida de la aplicación.

CAPÍTULO 5

EXPERIMENTACIÓN

Una vez que se explica con detalle los diferentes módulos de los que está dotado la aplicación, así como los objetivos que se tratan de alcanzar con ellos, es necesario realizar un proceso de experimentación con el algoritmo para determinar si los objetivos se cumplen. De esta manera se puede determinar cuales son los límites del algoritmo y cuales son los trabajos futuros a realizar para mejorarlo. La experimentación con el algoritmo se dividide en dos bloques.

- Bloque 1: Estudio de situaciones concretas de interés para analizar el comportamiento del programa ante ellas. Se estudian diferentes situaciones en las que la detección es cada vez más compleja.

- Bloque 2: Estudio de un determinado número de vídeos, tanto en interiores como exteriores, tanto de día como de noche, que nos permitan obtener una serie de conclusiones (porcentaje de personas detectadas, porcentaje de objetos en movimiento no detectados, análisis de situaciones especiales,...).

5.1. Estudio de situaciones concretas de interés

En esta sección se analiza distintas situaciones desde menor complejidad hasta mayor complejidad.

Situación 1: Detección y seguimiento de una única persona

Como se comenta en el apartado 4.4 al tratarse de una sola persona podemos hablar no solo de detección en cada frame sino también de seguimiento. Teniendo en cuenta todas las pruebas de campo que se realizan con esta situación se puede concluir que el comportamiento de la aplicación es óptimo. Tanto en interiores como exteriores se consigue un seguimiento de la persona (detección de la persona en cada frame), salvo rara excepción, durante el 100 % del tiempo. Cabe indicar que estas situaciones en las que no se detecta a la persona correctamente son sobretodo en exteriores ya que en algunas ocasiones no se fija a la persona exacta, sino que algunas veces la superficie es mayor debido a ruidos y algunas veces es menor debido a que el background se actualiza más rápidamente.

En la figura 5.1 se muestran ejemplos del funcionamiento del programa. Como se puede ver en los dos primeros casos la respuesta es óptima ante entornos interiores, realizando la detección de la persona en cada frame de manera perfecta durante todo el recorrido y consiguiendo que la caja englobante (del inglés convex hull) envuelva a la persona en su totalidad. En cambio los otros ejemplos muestran algunos de los problemas que presenta el algoritmo en algunas ocasiones. En el caso número 3 se puede observar como en exteriores el funcionamiento de la aplicación en muchos casos es peor, no representando a la persona en su totalidad o creando falsos positivos debidos a ruido provocado por la iluminación o por sombras. En el caso número 4 se puede observar como debido a que la camiseta del sujeto es muy parecida al fondo de la escena este se actualiza muy rápidamente provocando que se corte la cabeza por el cuello dejando de detectarla. Por último en el caso número 5 se puede observar como el tamaño del blob es mayor que el de la persona debido a la sombra que aparece en el suelo.

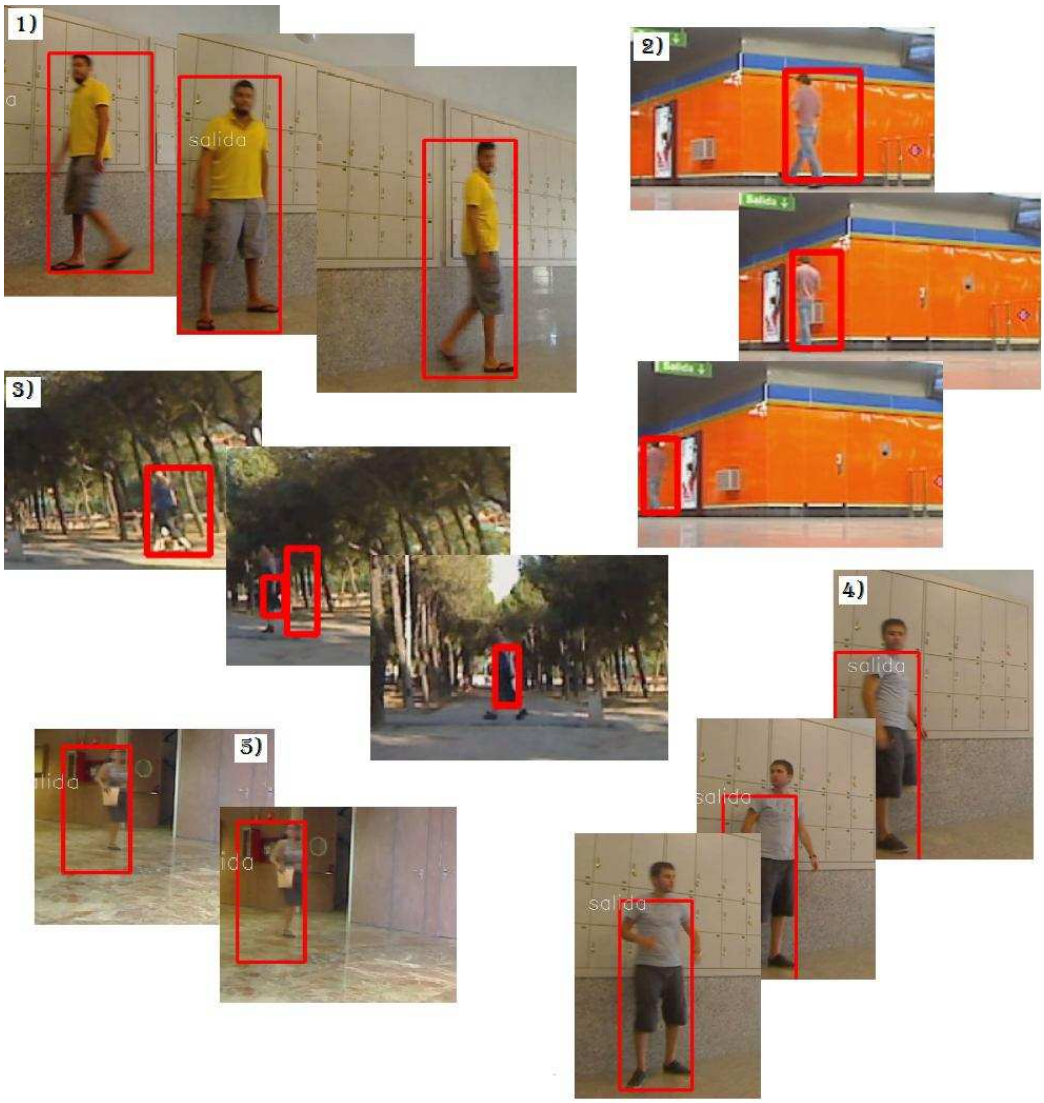

Figura 5.1: Detección y seguimiento de una persona.

Una vez se analiza el rendimiento ante esta situación se muestra el foreground en distintas casos para entender mejor lo que se está explicando.

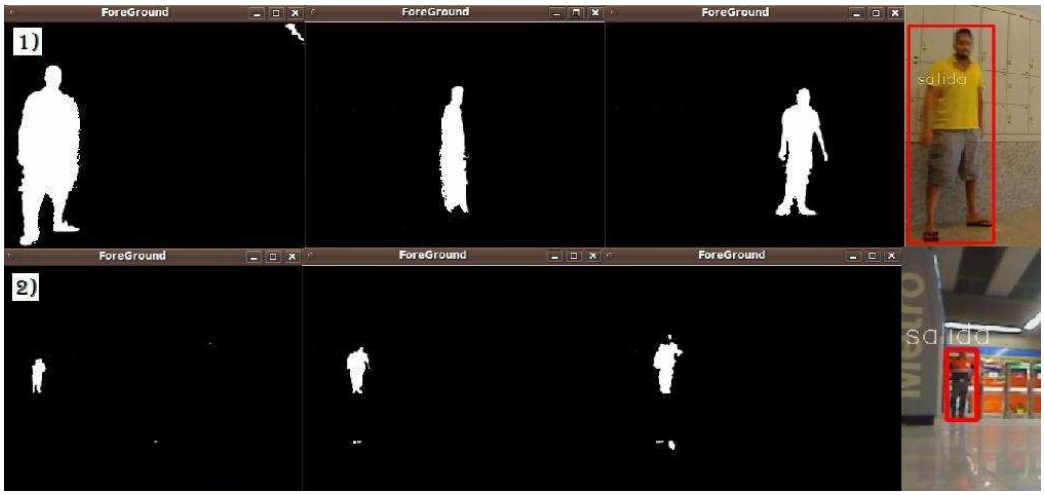

Figura 5.2: Foreground en la detección de una persona.

En el primer caso se puede observar como el foreground representa a la persona en cada frame de manera correcta con lo que se consigue una detección en cada uno de ellos perfecta. En el segundo caso vemos como se le corta la cabeza al individuo debido a que su camisa naranja empieza a coincidir con el fondo naranja del metro.

Por tanto, como conclusión, se puede verificar que ante la detección y seguimiento de una persona, especialmente interesante para la interacción robot-humano, el algoritmo tiene unas prestaciones óptimas y válidas.

Situación 2: Detección de dos o más personas

Evidentemente a medida que aumenta el número de sujetos en la zona de acción el rendimiento de la aplicación disminuye, si bien sigue teniendo unos resultados más que aceptables. Mientras las personas se mueven por separado en la zona de influencia el rendimiento del algoritmo es igual al que se comenta para el caso de una sola persona (ejemplos 1 y 3 en la figura 5.3), es decir, se detecta a las personas en el 100 % de los frames como normal general.

La aplicación en muy raras ocasiones empieza a tener problemas (si se trabaja desde vídeo y se graba otro vídeo, no en tiempo real) de coste computacional cuando el número de personas es muy elevado, provocando ralentizaciones que dan lugar a dobles figuras en el foreground si el procesador

no es lo suficientemente rápido. En el ejemplo 2 de la figura 5.3 se puede observar como en está ocasión para tres personas en una de ellas se produce esto (la doble silueta está indicada mediante un rectángulo azul, al igual que el aumento del blob debido a esto). Cabe destacar que esta situación no es la normal tampoco, por ejemplo en el vídeo 1 y 2 que se comenta en el siguiente apartado (5.2) tienen lugar situaciones parecidas con más personas en las que no se produce esto.

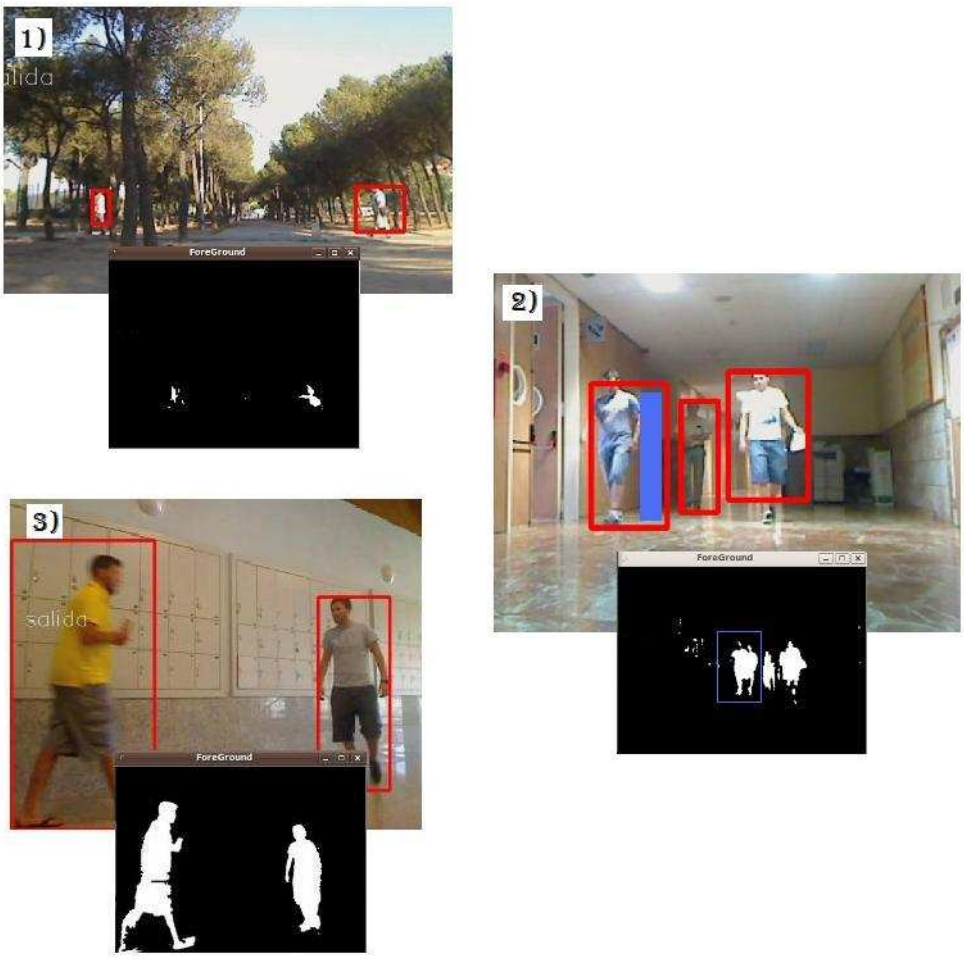

Figura 5.3: Detección de dos personas o más por separado.

Un problema importante de difícil solución para este caso es cuando aparecen grupos de personas unidas, sobretodo cuanto mayor es la distancia. En este caso en el foreground se detecta como un solo grupo de píxeles blancos

adyacentes. Esto no solo provoca que sea imposible detectar a cada persona por separado, sino que además si se es muy restrictivo en los parámetros utilizados para filtrar objetos ni se detecte al grupo de personas (ver caso 1 en la figura 5.4).

Otro problema a analizar cuando hay más de una persona es el cruce entre ellas. En este caso el algoritmo de segmentación detecta una único grupo de píxeles blancos, y además por lo general con mucho ruido. Este grupo de píxeles se asigna como un blob y si supera la fase de filtrado de objetos se asume que se detecta una persona, pero con una forma que es la superposición de las 2, y en todo caso solo se está detectando a una de las dos personas (ver caso 2 en la figura 5.4).

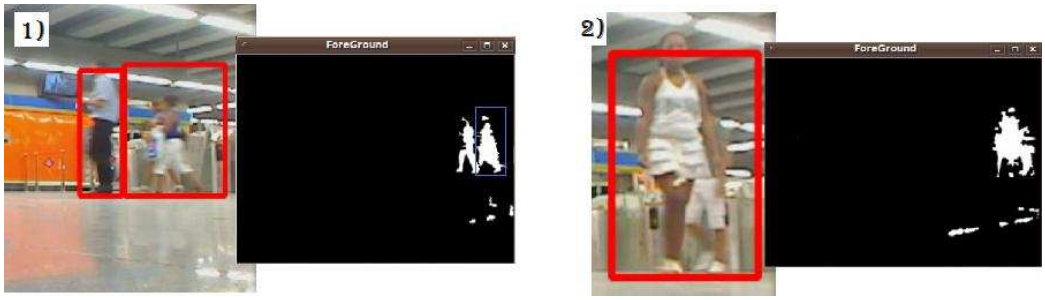

Figura 5.4: Problemas en la detección de dos personas o más.

Situación 3: Filtrado entre objetos y personas

En este apartado se analiza cuanto de veraz es el algoritmo a la hora de discriminar objetos respecto a personas, es decir, se analiza la eficacia del módulo de detección de personas que se ve en el apartado 4.4. Como se puede ver en la figura 5.5, la respuesta del algoritmo es positiva para todos esos objetos cuando se quiere detectar una persona en un entorno favorable. El inconveniente surge cuando, como se ve en los apartados anteriores, es necesario ampliar los rangos de filtrado bien sea por que el entorno es exterior o con mala iluminación y por tanto las personas no están tan bien definidas en el foreground, porque hay muchos objetos en el fondo de manera que la persona pasá por detrás quedando representada de cintura para arriba por ejemplo, o porque como se comenta en el apartado anterior aparecen varías personas unidas de forma que se detecta como un solo blob el cual se quiere representar. Debido a esto, si por ejemplo se aumenta la relación ancho/largo hasta 1 para mejorar estos casos, se empieza a detectar el balón de baloncesto.

El resto de los objetos presentes en la figura 5.5 se siguen filtrando. Por tanto, dependiendo de la situación y de los objetivos que se pretendan alcanzar con el algoritmo en cada caso, se debe determinar los rangos de filtrado, cuanto más generosos más casos de personas se detectan pero por el contrario más falsos positivos hay.

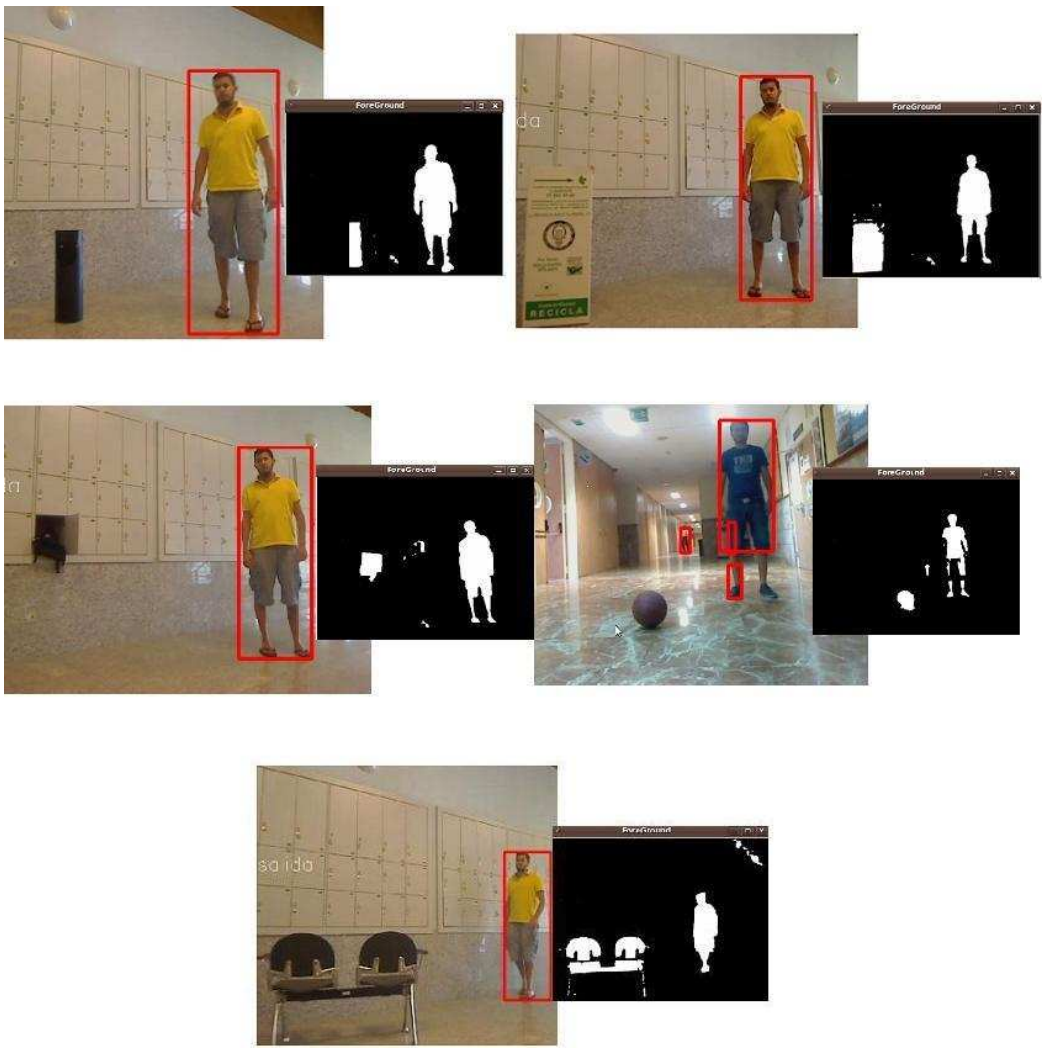

Figura 5.5: Ejemplo de filtrado de objetos.

Situación 4: Detección de personas estáticas

Es muy probable que a estas alturas el lector se pregunte que ocurre si una persona aparece en el campo de acción y permanece estática en una

posición durante un largo periodo de tiempo. Teniendo en cuenta la teoría del algoritmo de segmentación basado en el modelo de Bayes que se desarrolla en el subapartado 4.2.1 se supone que a medida que pase el tiempo el background se va actualizando introduciendo a la persona estática en el mismo hasta llegar un punto en que la persona forme parte totalmente del background y por tanto se deja de detectar. En este subapartado se realiza un estudio sobre esta situación con el objetivo de confirmar la hipótesis descrita y analizar los tiempos en los que se deja de detectar la persona.

Como se puede observar en el ejemplo de la figura 5.6 ocurre lo anteriormente mencionado. Desde el segundo dieciocho ya no se detecta la persona de manera correcta sino que por el contrario se fracciona en dos al actualizar más rápidamente el centro de la silueta de la persona. Esto provoca que, o bien la aplicación determine que hay dos personas en la escena de mitad de tamaño o si se es más restrictivo en el filtrado de objetos que ni siquiera detecte que hay una. En este mismo tiempo se observa como ya el resultado obtenido en el foreground es mucho peor si bien es a partir del minuto y medio cuando ya se empieza a observar casi la totalidad de la persona en el background mientras que en el foreground solo queda parte de la cara.

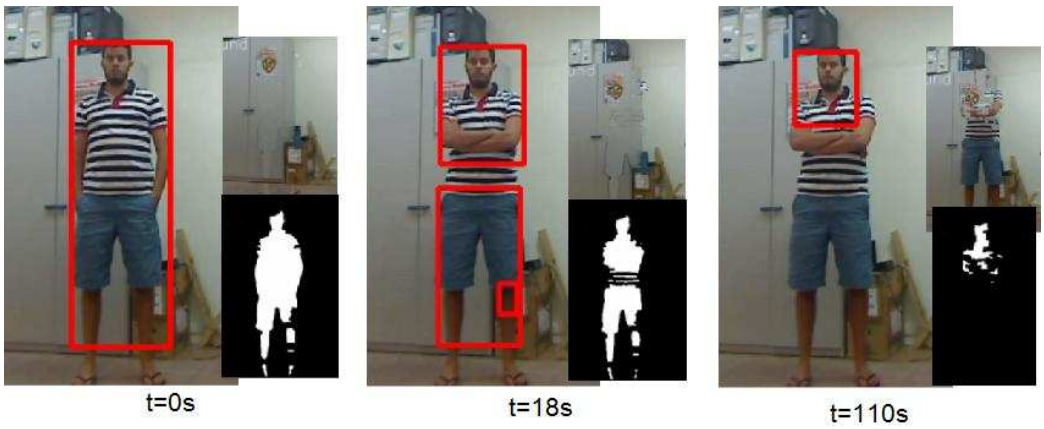

Figura 5.6: Evolución de una persona estática.

Indicar que los tiempos en los que la detección deja de ser correcta o deja de aparecer la persona por completo en el primer plano no son siempre los mismos, dependiendo entre otras cosas de los parámetros elegidos para el algoritmo o de la relación entre los colores del fondo y la persona.

5.2. Estudio de secuencias de vídeo

Como se comenta al principio de este capítulo en esta sección se realiza el estudio de un determinado número de vídeos en diversas circunstancias que nos permiten obtener una serie de conclusiones. Para mostrar los resultados de la forma más clara posible cada vídeo está estructurado en 4 partes. La primera de ellas es una tabla donde se hace una descripción de la escena y se colocan las características más relevantes que se tienen en cuenta a la hora de realizar la grabación. La segunda de las partes es una imagen del escenario donde se graba el vídeo para poder ver y entender mejor las principales características comentadas en la tabla anterior. La siguiente parte es una tabla donde se muestran los principales resultados que se obtienen del estudio de la secuencia de vídeo. Por último se incluye una parte donde se analizan de forma minuciosa los resultados anteriormente mostrados y se comentan las situaciones más relevantes.

5.2.1. Vídeo 1: Estación de metro de San Nicasio I

En primer lugar se estudia el comportamiento de la aplicación en interiores, comenzando por un vídeo que es grabado en el interior de la parada de metro de San Nicasio, Leganés. A continuación se muestran las principales características de esta prueba en la tabla 5.1.

<table>
<tr><th colspan="2">Vídeo 1: Estación de metro de San Nicasio I.</th></tr>
<tr><td colspan="2">Descripción: El vídeo es grabado en el interior de la parada de metro de San Nicasio, enfocando la cámara hacia el punto donde las personas atraviesan las barreras para entrar o salir del metro. Como se puede apreciar en la figura 5.7 se trata de un sitio amplio, donde el flujo de personas es elevado en momentos puntuales (cuando llegan los metros).</td></tr>
<tr><td>Hora: 19.50 h</td><td>Duración: 19:47 min</td></tr>
<tr><td>Iluminación: Artificial, de los fluorescentes situados en el techo de la estación.</td><td>Altura a la que se coloca la cámara: La cámara se situa en el suelo.</td></tr>
<tr><td>Parámetros del algoritmo de segmentación:
Lc= 64
Lcc= 32
Alpha 2= 0.005</td><td>Parámetros de algoritmo de filtrado de personas:
200 píxeles< Área < 15.000 píxeles
0,5 <= Ratio <= 0,85
0,1 <= Rec <= 0,5</td></tr>
<tr><td>Personas que aparecen en el vídeo: 56</td><td>Objetos en movimiento que aparecen en el vídeo: 21</td></tr>
</table>

Cuadro 5.1: Características del vídeo 1.

Como se puede observar los valores utilizados para el algoritmo de segmentación son los indicados para ambientes interiores en el apartado 4.2.3. Para el algoritmo de filtrado de objetos se filtran áreas menores a 200 píxeles, este valor no puede ser mayor debido a que se quiere detectar personas que están a una distancia considerable y por tanto su área es pequeña. Por contra solo se permite áreas de hasta 15000 píxeles debido a que las personas pasan a una cierta distancia. Tanto para el ratio como para la relación ancho/alto se tienen en cuenta los valores medios mencionados para personas en el apartado 4.4 (en el ratio 0.7 y en la relación ancho/alto 0.3). Indicar que

el número de objetos en movimiento es tan elevado debido al reflejo de las personas en el suelo que en ocasiones detecta el algoritmo de segmentación.

A continuación se muestra el background inicial en la figura 5.7.

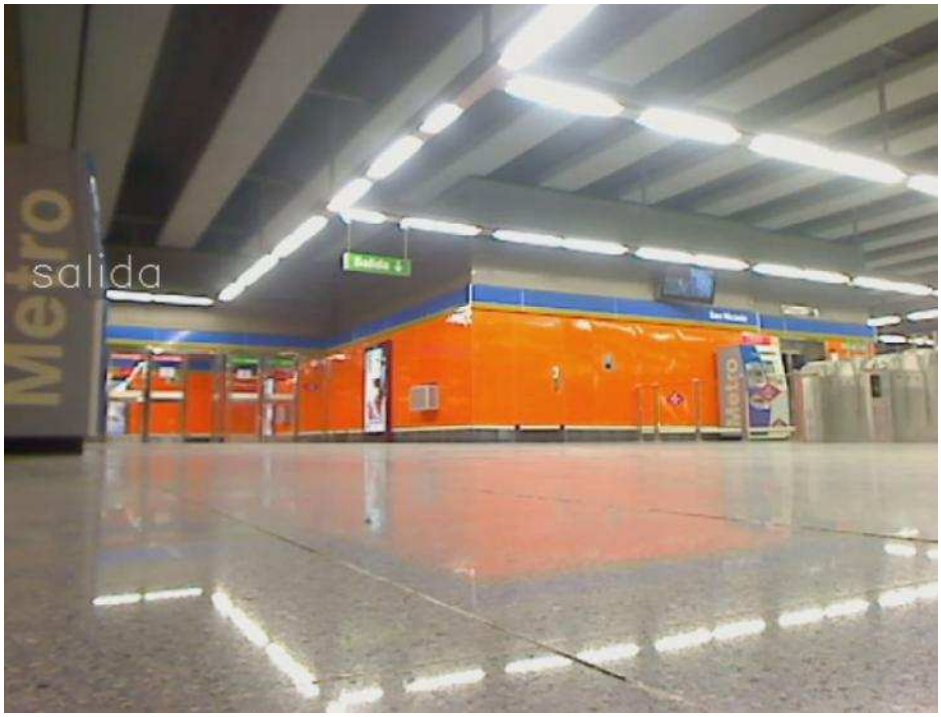

Figura 5.7: Escena donde tiene lugar la grabación del vídeo 1.

Los resultado que se obtienen son:

Total de personas detectadas en algún momento:	52/56
Personas detectadas el 100 % del tiempo:	16/52
Personas detectadas entre el 80 % y el 100 % del tiempo:	18/52
Personas detectadas entre el 60 % y el 80 % del tiempo:	14/52
Personas detectadas un tiempo inferior al 60 %:	4/52
Detecciones falsas de objetos:	1/21

Cuadro 5.2: Resultados obtenidos con el vídeo 1.

Se observa como los resultados que se obtienen no son tan exitosos como se podía esperar en un principio. Analizando el vídeo se comprueba que esto

se debe a lo restrictivo que son los parámetros para el filtrado de objetos. Los valores medios propuesto en 4.4 son adecuados cuando nos encontramos con un entorno en el que solo aparecen personas por separado y poco ruido (caso 1 y 2 en la figura 5.8), pero cuando aparecen grupos de personas o personas con formas extrañas y los problemas que esto provoca, los cuales se mencionan en 5.1, se deja de aceptar muchas situaciones que se desearían mostrar, ya que es mejor saber que en un lugar donde hay dos personas hay una que ninguna por ejemplo (ver casos 3 y 4 en la figura 5.8 para observar situaciones donde la detección no es correcta). Por el contrario gracias a ser tan restrictivos solo hay un falso positivo.

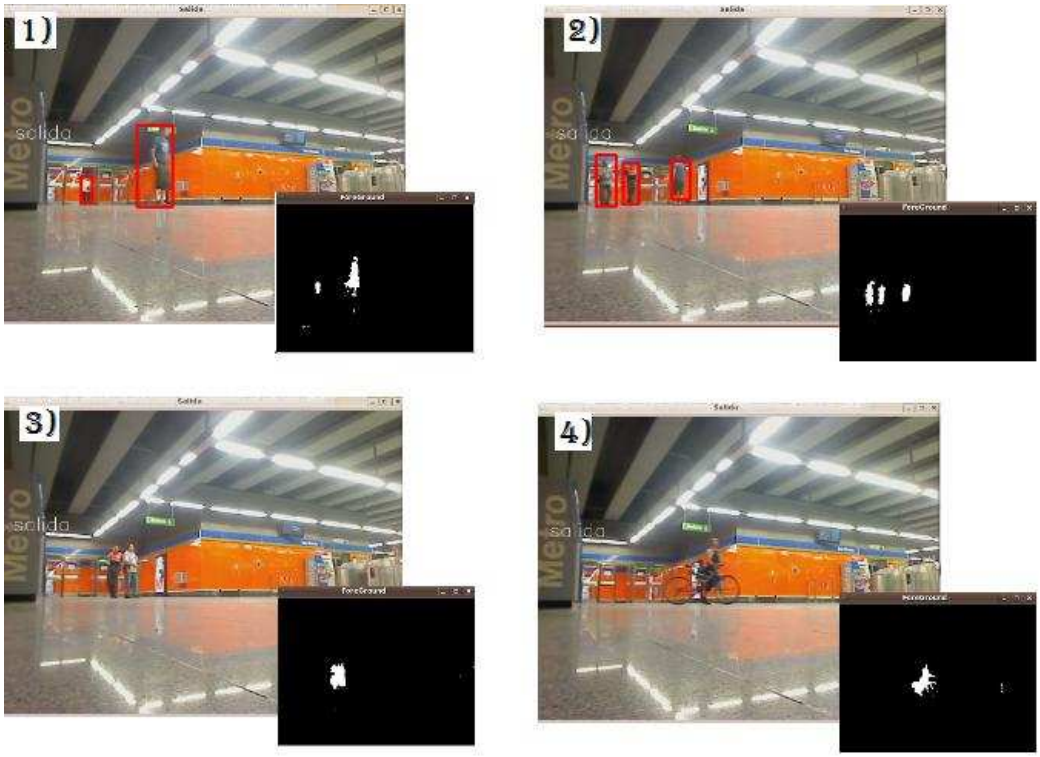

Figura 5.8: Situaciones con parámetros de filtrado restrictivos.

5.2.2. Vídeo 2: Estación de metro de San Nicasio II

En este segundo vídeo se quiere analizar el mismo entorno interior con flujo de personas elevado en momentos puntuales pero utilizando unos valores para filtrar objetos en movimiento más permisivos. Por tanto se espera obtener unos mejores resultados en este tipo de entorno.

<table>
<tr><th colspan="2">Vídeo 2: Estación de metro de San Nicasio II.</th></tr>
<tr><td colspan="2">Descripción: El vídeo es grabado en el interior de la parada de metro de San Nicasio al igual que el vídeo anterior. Lo que se pretende con está secuencia es estudiar la misma situación anterior pero utilizando en la misma unos parámetros menos restrictivos para así intentar aumentar lo máximo posible el número de personas que son detectadas un 100 %.</td></tr>
<tr><td>Hora: 19.30 h</td><td>Duración: 5:04 min</td></tr>
<tr><td>Iluminación: Artificial.</td><td>Altura a la que se coloca la cámara: La cámara se situa en el suelo.</td></tr>
<tr><td>Parámetros del algoritmo de segmentación:
Lc= 64
Lcc= 32
Alpha 2= 0.005</td><td>Parámetros de algoritmo de filtrado de personas:
200 píxeles < Área < 15.000 píxeles
0.4 <= Ratio <= 1
0,1 <= Rec <= 2</td></tr>
<tr><td>Personas que aparecen en el vídeo: 29</td><td>Objetos en movimiento que aparecen en el vídeo: 7</td></tr>
</table>

Cuadro 5.3: Características del vídeo 2.

Los valores para el algoritmo de segmentación se mantienen respecto al vídeo 1 pero como se puede observar para el módulo de detección únicamente de personas el ratio admitido pasa de entre 0.5-0.85 a 0.4-1 y la relación ancho/alto de entre 0.1-0.5 a 0.1-2. El fondo inicial de este vídeo es el mismo que el del vídeo 1 (figura 5.7).

Los resultados que se obtienen en este caso son:

Total de personas detectadas en algún momento:	27/29
Personas detectadas el 100 % del tiempo:	14/27
Personas detectadas entre el 80 % y el 100 % del tiempo:	11/27
Personas detectadas entre el 60 % y el 80 % del tiempo:	2/27
Personas detectadas un tiempo inferior al 60 %:	0
Detecciones falsas de objetos:	3/7

Cuadro 5.4: Resultados obtenidos con el vídeo 2.

Como se puede observar en los resultados, el cambio de los parámetros respecto al vídeo 1 provoca que se consigan unos mejores resultados en cuanto a la detección de personas. Por ejemplo, en el vídeo 1 se detecta todo el tiempo el 31 % de las personas mientras que ahora el 52 %. Cabe señalar que cuando se trata de personas individuales que no se cruzan el éxito es del 100 %, como en el vídeo anterior, pero en este caso gracias a ser más permisivos en el filtrado de objetos se consigue detectar a más personas en situaciones extrañas como con exceso de ruido, grupos de personas o cruces. Otro dato esclarecedor es que ahora ninguna persona es detectada menos del 60 % del tiempo.

Destacar que las 2 personas que no son detectadas en ningún intervalo de tiempo se debe a que desde el inicio están sentadas detrás de la puerta de acceso al metro, la cual se encuentra a una gran distancia, de manera que solo se detecta la mitad de sus cuerpos y por ello no se clasifican como personas.

Al cambiar los parámetros, como se comenta en la situación 3 del apartado 5.1, aumenta el número de falsos positivos como era de esperar. Por un lado aparecen dos reflejos que ahora son detectados como personas (un ejemplo de sombra que si pasa el filtro en este caso y no lo hubiera pasado en el anterior se puede ver en la figura 5.9) y por otro se muestra en un determinado momento la pantalla de televisión al cambiar bruscamente de color, problema que no ocurre en el primer vídeo. Por ello la relación ancho/altura en estos entornos se aconseja bajarla a 1 pues se siguen detectando los grupos de personas pero problemas como el de la tele no aparece.

Figura 5.9: Detección de la sombra.

Por último no se quiere desaprovechar la oportunidad que brinda este vídeo de mostrar el potencial de esta aplicación, mostrando en la figura 5.10 un momento del vídeo en el que el flujo de personas en escena es muy elevado. También se puede observar en la misma dos falsos positivos debido a los reflejos así como la detección de dos personas como una debido a que van unidas.

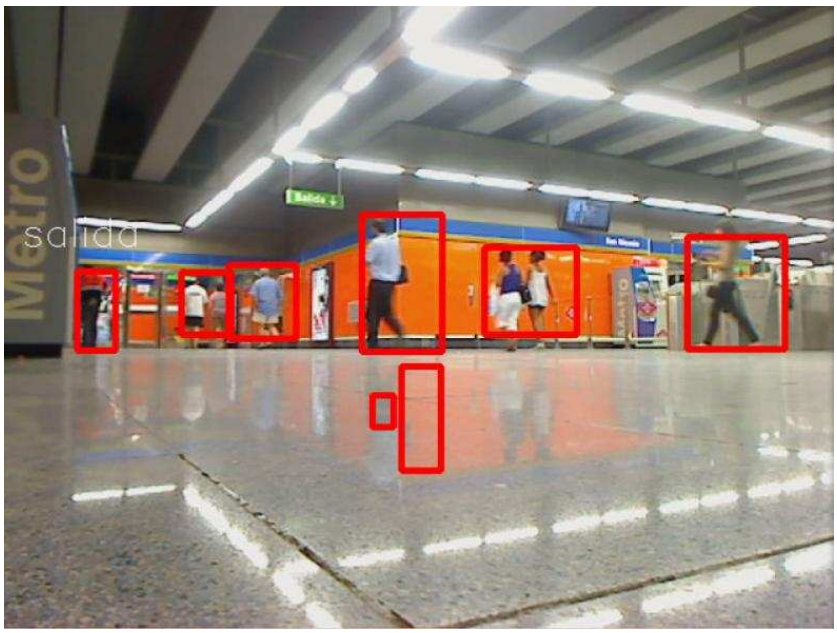

Figura 5.10: Detección de un número elevado de personas a la vez.

5.2.3. Vídeo 3: Segunda planta del Edificio Bethancourt

En este vídeo se sigue analizando la respuesta de la aplicación en entornos interiores. En este caso el lugar a estudiar es el interior de la Universidad Carlos III de Madrid. A continuación se muestran las principales características de esta prueba de campo en la tabla 5.5.

<table>
<tr><th colspan="2">Vídeo 3: Segunda planta del Edificio Bethancourt</th></tr>
<tr><td colspan="2">Descripción: El vídeo es grabado en la segunda planta del Edificio Bethancourt de la Universidad Carlos III en el mes de julio por lo que el tránsito de personas no es elevado al haber acabado el curso. En la figura 5.11 se puede apreciar de forma más clara la escena donde tiene lugar la grabación. Como se puede observar en ella se trata de la intersección de dos pasillos, uno principal donde se encuentran situados los despachos de los profesores y uno secundario para acceder a las aulas de informática.</td></tr>
<tr><td>Hora: 12.00 h</td><td>Duración: 10:37 min</td></tr>
<tr><td>Iluminación: Solar y artificial, la natural a través de grandes ventanales y la artificial por medio de fluorescentes situados en el techo del pasillo</td><td>Altura a la que se coloca la cámara: A un metro del suelo sobre una mesa.</td></tr>
<tr><td>Parámetros del algoritmo de segmentación:
Lc= 64
Lcc= 32
Alpha 2= 0.005</td><td>Parámetros de algoritmo de filtrado de personas:
1000 píxeles < Área < 25.000 píxeles
0,40 <= Ratio <= 0,85
0,1 <= Rec <= 0.8</td></tr>
<tr><td>Personas que aparecen en el vídeo: 16</td><td>Objetos en movimiento que aparecen en el vídeo: 2</td></tr>
</table>

Cuadro 5.5: Características del vídeo 3.

También en este caso los mejores parámetros para el algoritmo de separación de BG y FG son los indicados para interiores en 4.2.3. Para el algoritmo de filtrado de objetos se filtran áreas menores a 1000 píxeles, que es un valor elevado, debido a que la distancia más lejana en la que puede aparecer una persona no es elevada (esto beneficia en que todas las detecciones pequeñas

debido a ruido se filtran). Por contra se permite hasta áreas de 25000 píxeles debido a que pueden aparecer personas muy próximas a la cámara. Para los valores del ratio y ancho/alto se utiliza la experiencia de los vídeos anteriores, sabiendo que en este caso por el lugar donde es, parece complicado que aparezcan grupos tan numerosos como pueden aparecer en el metro, por lo que se puede disminuir la máxima relación ancho/alto.

A continuación se muestra el background inicial en la figura 5.11.

Figura 5.11: Escena donde tiene lugar la grabación del vídeo 3.

Los resultados que se obtienen para esta secuencia de vídeo son:

Total de personas detectadas en algún momento:	16/16
Personas detectadas el 100 % del tiempo:	15/16
Personas detectadas entre el 80 % y el 100 % del tiempo:	0
Personas detectadas entre el 60 % y el 80 % del tiempo:	1/16
Personas detectadas un tiempo inferior al 60 %:	0
Detecciones falsas de objetos:	0

Cuadro 5.6: Resultados obtenidos con el vídeo 3.

Como se puede apreciar en la tabla 5.2.3 los resultados que se obtienen son muy buenos ya que todas las personas que aparecen en él son detectadas y el 94 % lo son el 100 % del tiempo. Que estos resultados sean mejores que los de los dos vídeos anteriores se debe en gran medida a que en este caso no hay cruce de personas o tantos personas unidas (ver figura 5.12 para observar detecciones correctas en esta prueba). Sin embargo surge un pequeño problema, existe un momento en el que las personas 2 y 3 aparecen una detrás de la otra desde el ángulo donde se encuentra la cámara ya que andan muy próximas. En esta franja de tiempo la aplicación solo muestra una persona pues en el foreground se ve la forma completa de una persona distorsionada por un lado por parte de la persona que se encuentra detrás. Es por esto que existe una persona que solo se le detecta entre el 60 % y el 80 % del tiempo. Dicho problema se puede observar perfectamente en la figura 5.13.

Figura 5.12: Detecciones correctas en el vídeo 3.

Figura 5.13: Detección de una sola persona debido a superposición.

5.2.4. Vídeo 4: Tercera planta del Edificio Bethancourt

Se continúa con el estudio del comportamiento en interiores de la aplicación. En este caso en otro lugar de la Universidad Carlos III de Madrid.

<table>
<tr><th colspan="2">Vídeo 4: Tercera planta del Edificio Bethancourt</th></tr>
<tr><td colspan="2">Descripción: El vídeo es grabado en la tercera planta del Edificio Bethancourt de la Universidad Carlos III y al igual que la secuencia anterior se graba en el mes de julio por lo que el tránsito de personas no es elevado. En la figura 5.14 se puede apreciar de forma más clara la escena donde tiene lugar la grabación. Como se puede observar en ella se trata de un pasillo de gran profundidad llena de despachos y laboratorios a los lados, de donde aparecen la mayoría de las personas.</td></tr>
<tr><td>Hora: 10.30 h</td><td>Duración: 5:58 min</td></tr>
<tr><td>Iluminación: En este caso sólo hay iluminación artificial por medio de fluorescentes situados en el techo del pasillo ya que a diferencia del caso anterior en esta parte del pasillo no se dispone de ventanales.</td><td>Altura a la que se coloca la cámara: La cámara se situa a medio metro del suelo sobre una silla.</td></tr>
<tr><td>Parámetros del algoritmo de segmentación:
Lc= 64
Lcc= 32
Alpha 2= 0.005</td><td>Parámetros de algoritmo de filtrado de personas:
300 píxeles < Área < 15.000 píxeles
0,4 <= Ratio <= 0,8
0,1 <= Rec <= 1</td></tr>
<tr><td>Personas que aparecen en el vídeo: 14</td><td>Objetos en movimiento que aparecen en el vídeo: 0</td></tr>
</table>

Cuadro 5.7: Características del vídeo 4.

En esta ocasión se continua utilizando los mismos valores para segmentación que en el resto de algoritmos interiores y se utiliza valores muy próximos a los utilizados en el vídeo 3 para el módulo de detección únicamente de personas, salvo en el área mínima que debido a la longitud del pasillo es bastante más baja.

En este caso el background inicial es:

Figura 5.14: Escena donde tiene lugar la grabación del vídeo 4.

Y los resultados que se obtienen para esta prueba de campo son:

Total de personas detectadas en algun momento:	13/14
Personas detectadas el 100 % del tiempo:	6/14
Personas detectadas entre el 80 % y el 100 % del tiempo:	4/14
Personas detectadas entre el 60 % y el 80 % del tiempo:	3/14
Personas detectadas un tiempo inferior al 60 %:	0
Detecciones falsas de objetos:	0

Cuadro 5.8: Resultados obtenidos con el vídeo 4.

Como se puede observar en este caso el 100 % de las personas no son detectadas y solo el 43 % de las detectadas lo son el 100 % del tiempo. Esto es debido a dos aspectos importantes a resaltar. En primer lugar, como se puede apreciar en la figura 5.15, al empezar a capturar la secuencia de vídeo hay 3 personas que se encuentran quietas en la escena por lo que desde un principio pasaron a formar parte del background y no son detectadas hasta que se mueven, por lo que no son detectadas el 100 % del tiempo.

Figura 5.15: Primera imagen de la secuencia de vídeo, se observa como las 3 personas que aparecen en la imagen forman parte del background.

El otro punto a resaltar es que hay una persona que ni se detecta y otras que no lo son en determinados intervalos debido a la profundidad del pasillo, ya que las personas que aparecen al final de este van a tener un área muy pequeña que no satisface las restricciones de área impuestas. En la imagen 5.16 se puede observar un ejemplo de esta situación. En caso de querer detectar las personas a esta distancia se debe disminuir el área mínima si bien aumentará el ruido considerablemente.

Figura 5.16: No detección de la persona por mótivo de restricción de área.

5.2.5. Vídeo 5: Laboratorio del departamento de Robótica y Automatización

Dado que uno de los principales objetivos para el que se desarrolla esta aplicación es la interacción robot-humano, buscando facilitar la comunicación entre ambos, en este último vídeo en interior se estudia el comportamiento del algoritmo ante la presencia de un ser humano a una distancia corta realizando distintos movimientos, intentando simular que información podría recibir el robot de este a través de esta aplicación. En la tabla 5.9 se muestran las principales características del vídeo.

<table>
<tr><th colspan="2">Vídeo 5: Laboratorio del departamento de Robótica y Automatización</th></tr>
<tr><td colspan="2">Descripción: El vídeo es grabado en el interior del laboratorio del departamento de Robótica y Automatización. Se graba con el laboratorio cerrado de manera que solo aparece y se mueve una sola persona en la zona de visión, con el objetivo de simular el funcionamiento de la aplicación en un robot para interactuar con un ser humano.</td></tr>
<tr><td>Hora: 16.30 h</td><td>Duración: 8:03 min</td></tr>
<tr><td>Iluminación: Artificial proporcionada por medio de los fluorescentes situados en el techo. Ventanas con estores a media altura.</td><td>Altura a la que se coloca la cámara: La cámara se situa 1 metro sobre el nivel del suelo</td></tr>
<tr><td>Parámetros del algoritmo de segmentación:
Lc= 64
Lcc= 32
Alpha 2= 0.01</td><td>Parámetros de algoritmo de filtrado de personas:
200 píxeles < Área < 30.000 píxeles
0,5 <= Ratio <= 0,9
0,1 <= Rec <= 0.9</td></tr>
<tr><td>Personas que aparecen en el vídeo: 1</td><td>Objetos en movimiento que aparecen en el vídeo: 0</td></tr>
</table>

Cuadro 5.9: Características del vídeo 5.

Señalar que respecto a los otros casos en interior se aumenta el valor de α_2 hasta 0.1 pues se consiguen mejores resultados. En cuanto al resto de valores señalar que como se puede observar se es bastante restrictivo en el

ratio y en la relación ancho/alto algo menos para poder detectar a la persona sentada por ejemplo. Por último dado que la persona está cerca de la cámara se permiten valores de área elevados.

El background inicial para este vídeo es:

Figura 5.17: Escena donde tiene lugar la grabación del vídeo 5.

En este vídeo no se adjunta la tabla con los resultados obtenidos dado la peculiaridad del mismo. Por tanto se procede directamente al análisis de los resultados. Durante los 8 minutos que dura aproximadamente el vídeo siempre que el sujeto aparece y se mueve por la zona de visión el algoritmo funciona perfectamente, incluso cuando permanece estático en un lugar pues es durante pocos segundos. Gracias al elevado ajuste que se puede hacer de las restricciones pues solo se quería detectar a una persona en un rango de distancia corto (eso si en cualquier posición) las falsas detecciones producidas por ruido son casi nulas. A continuación se muestra en la figura 5.18 el correcto funcionamiento del algoritmo ante distintas posiciones del individuo así como un falsa detección del balón de baloncesto debido a que está partido por la mitad al aparecer detrás de la pata de una mesa. Esto se puede solucionar aumentando el valor inferior que delimita las áreas válidas pues

evidentemente en ese lugar no va a aparecer una persona de ese tamaño.

Figura 5.18: Situaciones a destacar en el vídeo 5.

5.2.6. Vídeo 6: Exteriores del metro de San Nicasio

A partir de este vídeo se comienza a analizar la respuesta de la aplicación en exteriores. En este primer vídeo se estudia el rendimiento del algoritmo en horario diurno. En la tabla 5.10 se muestran las principales características de la secuencia.

<table>
<tr><th colspan="2">Vídeo 6: Exterior metro de San Nicasio</th></tr>
<tr><td colspan="2">Descripción: El vídeo es grabado en el exterior de una de las salidas de la parada de metro de San Nicasio. Se trata de una zona sombreada rodeada de árboles de gran altura como se puede observar en la figura 5.19 y que se caracteriza por tener un elevado paso de transeúntes.</td></tr>
<tr><td>Hora: 19 h</td><td>Duración: 13:10 min</td></tr>
<tr><td>Iluminación: Solar y, dada la hora, lateral. Además debido a los árboles son numerosas las sombras.</td><td>Altura a la que se coloca la cámara: La cámara se situa a medio metro sobre el nivel del suelo.</td></tr>
<tr><td>Parámetros del algoritmo de segmentación:
Lc= 128
Lcc= 64
Alpha 2= 0.005</td><td>Parámetros de algoritmo de filtrado de personas:
200 píxeles < Área <15.000 píxeles
0.4 <= Ratio <= 0.9
0,1 <= Rec <= 1</td></tr>
<tr><td>Personas que aparecen en el vídeo: 32</td><td>Objetos en movimiento que aparecen en el vídeo: 45</td></tr>
</table>

Cuadro 5.10: Características del vídeo 6.

Los parámetros del algoritmo de extracción del foreground son los recomendados en 4.2.2 para ambientes exteriores salvo el valor de α_2, pues con este valor se obtienen mejores resultados. Para el módulo de distinción entre personas y el resto se utilizan valores que permitan detectar grupos de personas que aparezcan juntas o se crucen dado el elevado flujo de personas que hay en momentos puntuales. Destacar el elevado número de objetos en movimiento clasificados, en su mayoría debido a sombras y al movimiento de las ramas de los árboles por acción del viento.

El background inicial de esta prueba es:

Figura 5.19: Escena donde tiene lugar la grabación del vídeo 6.

Mientras que los resultados que se obtienen son:

Total de personas detectadas en algún momento:	32/32
Personas detectadas el 100 % del tiempo:	20/32
Personas detectadas entre el 80 % y el 100 % del tiempo:	6/32
Personas detectadas entre el 60 % y el 80 % del tiempo:	6/32
Personas detectadas un tiempo inferior al 60 %:	0
Detecciones falsas de objetos:	21/45

Cuadro 5.11: Resultados obtenidos con el vídeo 6.

En primer lugar destacar que los resultados, en cuanto a la detección en algun momento de personas, son mejor de lo esperado. Por el contrario se observa como la proporción de objetos en movimiento que no se filtran es elevada (entorno al 50 %). Como se explica en otros vídeos esto es debido a que el rango de filtrado se ha abierto para mejorar la detección de personas. Se puede corregir este defecto cambiando el algoritmo para que filtre las áreas pequeñas, pero entonces se deja de detectar a las personas que se encuentran

más lejos por lo que se cree conveniente no cambiarlo en esta situación. En la figura 5.20 se muestran ejemplos de detección correcta, mientras que en la imagen 5.21 se puede apreciar como se detecta una rama del árbol.

Figura 5.20: Ejemplo de detecciones correctas en el vídeo 6.

Figura 5.21: Detección de una rama.

En cuanto a las personas que son detectadas sólo un intervalo de tiempo comprendido entre el 60 % y 80 % del tiempo se debe a que el camino escogido por estas es en la misma dirección del enfoque de la cámara, de manera que a medida que la persona camina se va alejando de la escena y el área de estas en el foreground es cada vez menor hasta que no cumple la condición mínima de área para ser detectada.

5.2.7. Vídeo 7: Paseo Paquita Gallego.

Por último para terminar esta sección se estudia un video nocturno. En la siguiente tabla se presentan las principales características del mismo.

<table>
<tr><td colspan="2">Vídeo 7: Camino paralelo a la calle Río Nervión en Leganés.</td></tr>
<tr><td colspan="2">Descripción: El vídeo es grabado en el paseo peatonal Paquita Gallego, que se encuentra cercana a la Universidad Carlos III. La secuencia es tomada en el mes de julio durante la noche para probar la aplicación en unas condiciones diferentes a las dispuestas hasta ahora. En la figura 5.22 se puede apreciar de forma más clara la escena donde tiene lugar la grabación, así como la luminosidad de la que dispone el paseo. Como se puede observar en la imagen, la secuencia es tomada desde una perspectiva lateral para detectar y seguir a los transeúntes durante un tramo corto.</td></tr>
<tr><td>Hora: 22.36 h</td><td>Duración: 10:07 min</td></tr>
<tr><td>Iluminación: En este caso a diferencia de los 6 vídeos anteriores la luz natural no es solar, sino lunar. Por otro lado también se dispone de luz artificial, ésta se encuentra situada en farolas cada 5 metros a ambos lados del camino.</td><td>Altura a la que se coloca la cámara: La cámara se situa a medio metro del suelo sobre un banco de madera que se encuentra en el camino.</td></tr>
<tr><td>Parámetros del algoritmo de segmentación:
Lc= 128
Lcc= 64
Alpha 2= 0.01</td><td>Parámetros de algoritmo de filtrado de personas:
200 píxeles< Área < 15.000 píxeles
0,4 <= Ratio <= 0,9
0,1 <= Rec <= 1</td></tr>
<tr><td>Personas que aparecen en el vídeo: 15</td><td>Objetos en movimiento que aparecen en el vídeo: 34</td></tr>
</table>

Cuadro 5.12: Características del vídeo 7.

Como en el vídeo anterior los valores Lc y Lcc son los recomendados para exteriores, mientras que en este caso se utiliza un α_2 de 0.01 para que el background se actualize más rápido debido al gran ruido por sombras que

aparece. Los valores para el módulo de diferenciación son semejantes a los que se utilizan en los últimos vídeos. Como en el vídeo anterior, también en el exterior, se puede observar como el número de objetos en movimiento que aparecen (considerándolos a partir de un cierto tamaño pues de muy pequeñas dimensiones hay en todo momento) es muy elevado, en este caso sobretodo debido a las sombras que producen las farolas como se acaba de comentar.

En la siguiente imagen se observa el fondo estático de la experimentación:

Figura 5.22: Escena donde tiene lugar la grabación del vídeo 7.

Los resultados que se obtienen se muestran en la tabla 5.13.

Total de personas detectadas en algún momento:	14/15
Personas detectadas el 100 % del tiempo:	5/14
Personas detectadas entre el 80 % y el 100 % del tiempo:	1/14
Personas detectadas entre el 60 % y el 80 % del tiempo:	3/14
Personas detectadas un tiempo inferior al 60 %:	5/14
Detecciones falsas de objetos:	7/34

Cuadro 5.13: Resultados obtenidos con el vídeo 7.

En esta secuencia hay varios aspectos importantes que hay que resaltar. En primer lugar se observa como el hecho de que sea un vídeo en el exterior, sumado a la nocturnidad del mismo, provoca que los resultados obtenidos sean cuanto menos no del todo satisfactorios. Pues por ejemplo el número de personas detectadas menos del 80 % del tiempo asciende al 57 %. A parte de las características del entorno, otra motivo que ayuda a estos valores tan elevados es el ángulo en el que se situa la cámara. Al ser éste perpendicular a la dirección en la cual caminan las personas, al aparecer una pareja en la escena sólo es detectada la que más cerca se encuentra de la cámara, mientras que la otra se encuentra tapada en su mayor parte por la primera. En la figura 5.23 se muestran dos detecciones correctas, destacar la gran sombra que provoca la persona en el ejemplo de la izquierda la cual se filtra por no cumplir los requisitos de ancho/alto. En la imagen 5.24 se pueden apreciar la situación que anteriormente se comenta en la que aparecen dos personas una detrás de otra.

Figura 5.23: Personas detectadas correctamente en el vídeo 7.

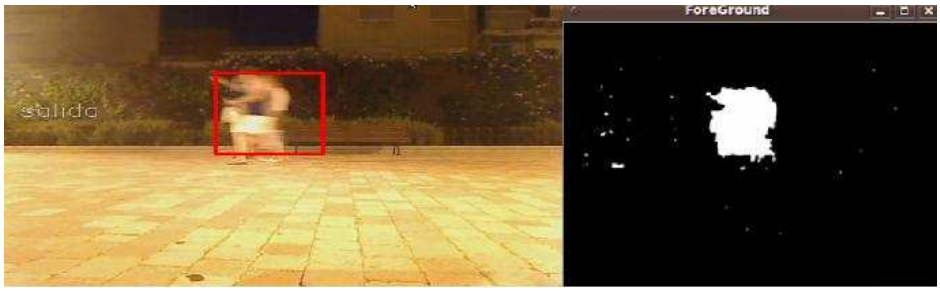

Figura 5.24: Persona no detectada por estar tapada por otra.

El segundo dato a resaltar es el número de detecciones falsas. Si bien es elevado, no lo es tanto en comparación a los objetos en movimiento detectados. Esto es debido a que gran cantidad de sombras son filtradas por no cumplir las restricciones de área, ratio o relación altura-ancho. Otra falsa detección es la de la ventana de una casa, en la cual se enciencde la luz. En las siguientes imágenes se puede observar de forma clara ambas situaciones.

Figura 5.25: Falsa detección, sombra.

Figura 5.26: Falsa detección, ventana.

Evidentemente en ambos casos el blob es de muy pequeño valor, por lo que si se aumenta un poco el valor mínimo de área de nuevo se resuelve el problema.

CAPÍTULO 6

CONLUSIONES

Este proyecto nace con el objetivo principal de desarrollar una herramienta para la detección de personas que se encuentran en el campo de visión a través de la extracción del foreground. Teniendo como objetivos secundarios el estudio teórico de la bibliografía existente, la experimentación con la aplicación desarrollada y la obtención de conclusiones y posibles trabajos futuros. Vistos los dos primeros es momento de obtener las conclusiones pertinentes.

En primer lugar el primer gran punto de interés de este proyecto es elegir cual es el algoritmo de segmentación más adecuado. De la bibliografía existente se destacan dos por su relación eficiencia-coste computacional: el FGD (basado en el modelo de Bayes) y el MOG (basado en el modelo de mezclas gaussianas). Una vez se analizan teóricamente ambos en la sección 4.2.1 se exponen las siguientes características en cuanto a prestaciones:

- Incorporación de background a la escena (objeto que pasado un tiempo se incorpora al fondo): Ambos algoritmos tienen un comportamiento óptimo. Con los parámetros preestablecidos en el MOG el tiempo es menor, pero en FGD este varía cambiando el parámetro alpha 2, pudiendo llegar a ser menor.

- Background dinámico (por ejemplo movimiento de hojas de un árbol): Correcto funcionamiento en FGD que no detecta como foreground este movimiento. Por el contrario en MOG en ocasiones si se detecta este background dinámico como foreground.
- Cambios de iluminación: En el FGD no se detectan las zonas afectas por el cambio de iluminación como foreground, mientras que en el MOG en ocasiones si.
- Vídeo comienza con foreground en escena: Ambos tienen comportamiento similar.
- Foreground estático (hace referencia a los blobs que por una razón u otra se encuentran parados en un momento dado): Ambos algoritmos presentan un comportamiento similar salvo el caso en el que una persona esta parada y realiza pequeños movimientos (por ejemplo una dependienta atendiendo a un cliente en una tienda), en este caso el comportamiento del MOG es mejor.
- Foreground dinámico y solitario:
  - Sin paradas: En este caso a distancias cortas y medias ambos algoritmos detectan el movimiento, si bien el algoritmo FGD representa la silueta de manera mucho más real, es decir, con mucho menor ruido. Por el contrario a largas distancias si la cámara no tiene una elevada resolución el algoritmo MOG presenta mejores prestaciones a la hora de detectar el movimiento.
  - Con paradas: Comportamiento similar en ambos casos salvo cuando el color del foreground es muy parecido al color del background en cuyo caso el MOG es algo mejor.
- Foreground dinámico y en grupo: Resultados similares en ambos casos.
- Foreground: personas que se cruzan: Comportamiento parecido también en ambos casos.

Una vez se realiza esta comparación entre ambos se selecciona el algoritmo FGD debido a que se busca obtener buenos resultados sobretodo a medias distancias. Con este algoritmo se consigue representar los contornos de manera mucho más fiel al tener menos ruido si bien se comporta peor en algunas situaciones concretas (personas lejanas, personas paradas desde un principio y personas con ropa semejante al fondo de la imagen).

El siguiente punto importante del algoritmo es el módulo de detección de blobs. Como se indica en el apartado 4.4 aquí se realiza el labelling de la imagen binaria del foreground, agrupando los píxeles blancos adyacentes en un blob.

A continuación se tiene que decidir que método utilizar para discriminar los objetos en movimiento que no son personas. Entre el método basado en contornos, el basado en regiones y el basado en las características del blob se opta por este último pues los dos primeros tienen problemas de complejidad y coste computacional o falta de desarrollo.

Finalizado este pequeño repaso sobre la aplicación es necesario ahora obtener una serie de conclusiones sobre la experimentación que se realiza.

Del primer apartado de la experimentación, estudio de situaciones concretas de interés (5.1) se pueden obtener las siguientes conclusiones. En primer lugar que el rendimiento de la aplicación a la hora de detectar a una única persona es casi inmejorable, consiguiendo tanto en entornos interiores como exteriores un seguimiento de la persona (detección de la persona en cada frame), salvo rara excepción, durante el 100 % del tiempo.

Cuando el número de personas es superior se consiguen resultados parecidos cuando estas se encuentran separadas, empezando a tener problemas cuando aparecen juntas o se cruzan ya que solo se representa a una de ellas. En cuanto a la discriminación de objetos en movimiento se puede asegurar que es un éxito cuando solo se quiere detectar a una persona mientras que el número de falsos positivos comienza a aumentar al ampliar el rango de filtrado para detectar situaciones más especiales (pareja de la mano, persona sentada,...).

Una vez analizado el primer módulo de la experimentación es necesario hacer lo mismo con el segundo. En primer lugar en la figura 6.1 se muestran los resultados obtenidos para entornos interiores siendo restrictivos con los parámetros de filtrado.

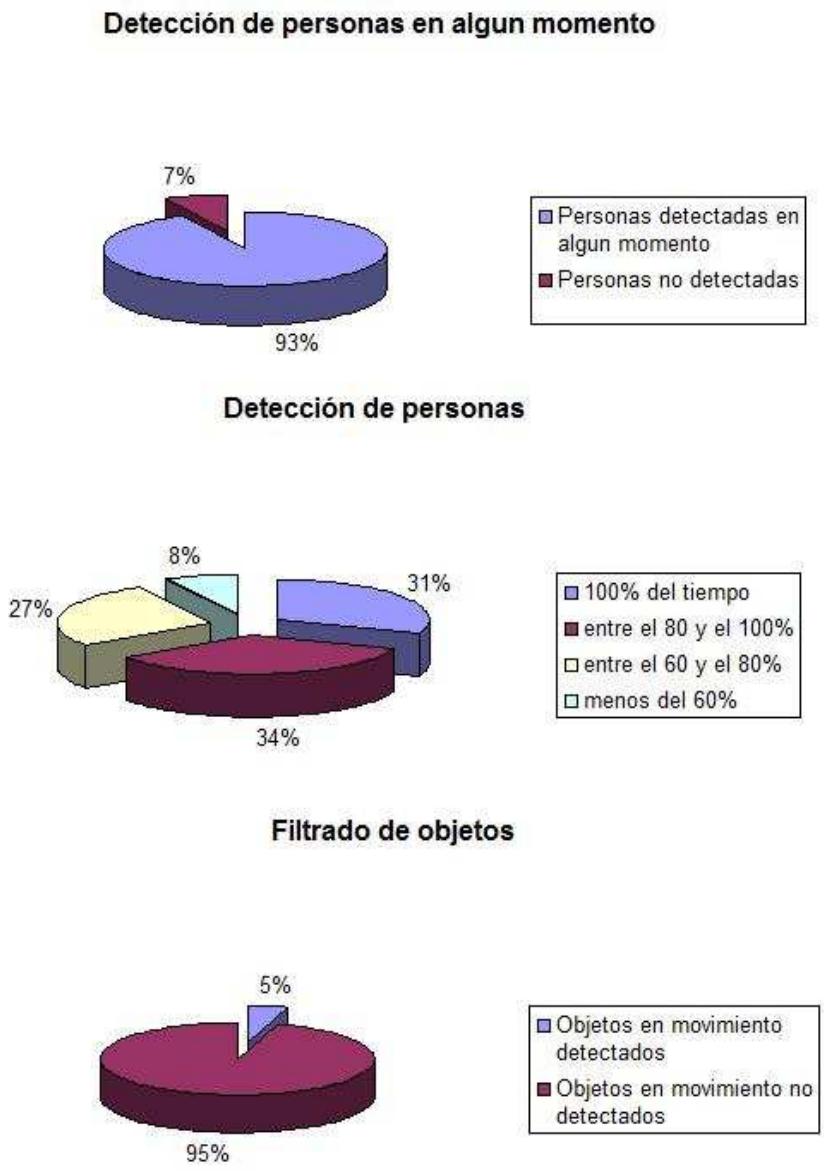

Figura 6.1: Resultados en vídeo interior con parámetros restrictivos.

Como se comenta en el apartado 5.2 se observa como los resultados obtenidos en la detección de personas no son tan óptimos como se esperaba, si bien se ha conseguido filtrar un gran número de objetos en movimiento. Al ampliar los rangos permitidos para mejorar la detección de personas los resultados obtenidos en interior son:

Figura 6.2: Resultados en vídeos interiores con parámetros para optimizar detección y seguimiento.

En la tabla 6.1 se observa como mejora el comportamiento del algoritmo en cuanto a personas si bien hay muchas más detecciones erróneas de objetos.

	Optimizar falsos positivos	Optimizar máxima detección de personas
Personas detectadas en algun momento	93 %	95 %
El 100 %	31 %	62 %
Entre el 80 y el 100 %	34 %	27 %
Entre el 60 y el 80 %	27 %	11 %
Menos del 60 %	8 %	0 %
Objetos detectados	5 %	43 %

Cuadro 6.1: Comparación de resultados obtenidos en interior.

Por último los resultados obtenidos en exteriores optimizando la detección de personas se muestran en la figura 6.3.

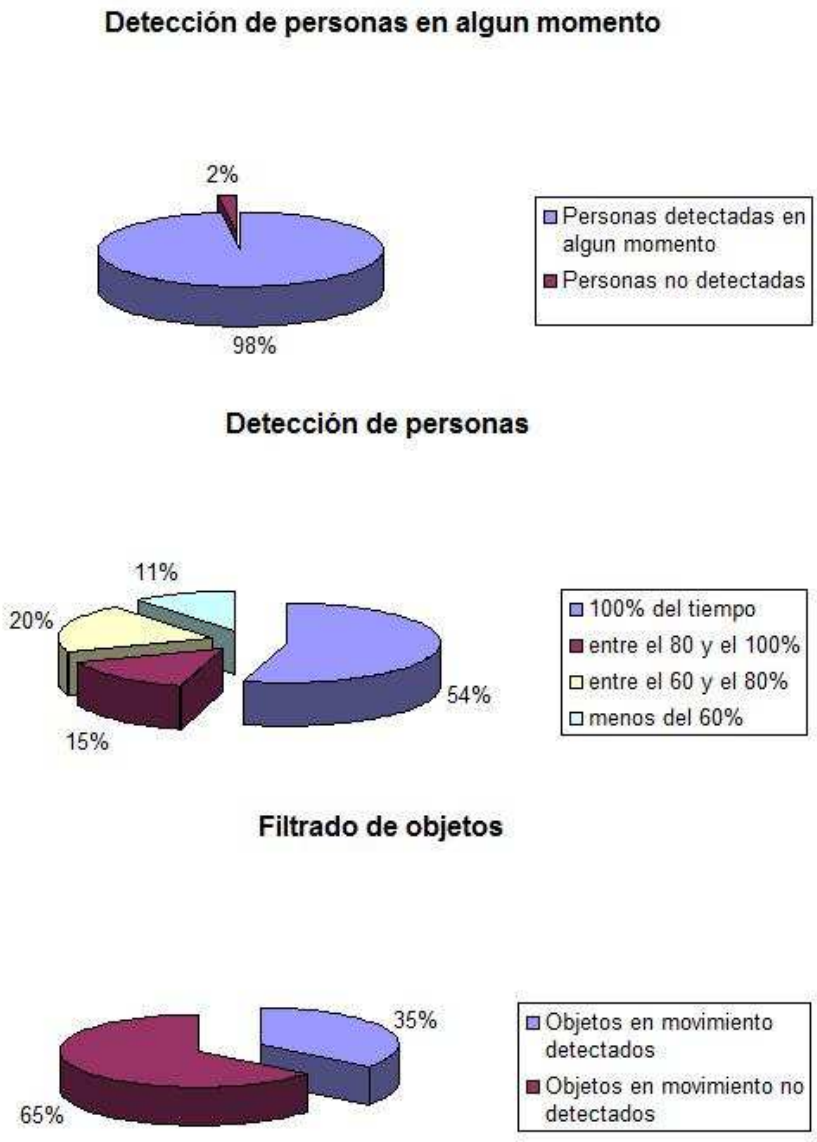

Figura 6.3: Resultados en vídeos exteriores.

En la tabla 6.2 se compara los resultados con los obtenidos para interior.

	Exterior	Interior
Personas detectadas	93 %	95 %
Seguidas el 100 %	54 %	62 %
Entre el 80 y el 100 %	15 %	27 %
Entre el 60 y el 80 %	20 %	11 %
Menos del 60 %	11 %	0 %
Objetos detectados	36 %	43 %

Cuadro 6.2: Comparación entre resultados obtenidos en interior y exterior.

Como se puede observar los resultados en exteriores son bastante peores que los que se obtienen en interior, tal y como era de esperar. Cabe resaltar que el porcentaje de personas detectadas es mayor en exterior debido a las circunstancias especiales que hay en los vídeos grabados en interior como personas estáticas desde el inicio o muy lejanas. Si estas pruebas de campo se extrapolan a una población mayor el porcentaje en interior sería mucho mayor. Por último mencionar que el porcentaje de objetos en movimiento detectados en interior es mayor debido a que en exteriores las sombras que se provocaban son tan grandes que en la gran mayoría son filtradas sin problema mientras que los reflejos en interiores tienen casi la misma forma que las personas.

CAPÍTULO 7

TRABAJO FUTURO

Es evidente que pese al buen rendimiento de la aplicación que se desarrolla en este proyecto, esto no supone una solución definitiva al problema de la detección de personas, ya que aún hay pequeños detalles que se pueden mejorar y así conseguir un comportamiento más fiable del programa desarrollado. Por otro lado a lo largo del desarrollo de esta aplicación se comprueba que es recomendable añadirle nuevos módulos para conseguir distintos objetivos que necesitan del paso previo de la detección pero que no forman parte de esta en sí. Es por esto que en un primer lugar se mencionan los trabajos futuros recomendados para mejorar la detección para seguidamente indicar posibles mejoras que se le pueden añadir a un programa de detección con el fin de obtener más y mejores resultados.

Para mejorar estos pequeños detalles se cree que las líneas de investigación futuras deben ir enfocadas en las siguientes direcciones.

En cuanto a la mejora de los algoritmos de detección se cree que hace falta una investigación profunda en el campo de la evaluación de algoritmos, ya que una correcta elección de los parámetros hace que el algoritmo mejore de manera sustancial. Se cree que es recomendable crear un sistema automático

para los distintos algoritmos que vaya probando con los distintos parámetros hasta dar con la solución óptima.

Otra línea de investigación futura es la de mejorar la relación eficiencia-coste computacional, bien sea depurando los algoritmos de segmentación o mejorando los procesadores de los ordenadores. Este es un punto clave al ser una aplicación en tiempo real, con un procesador de mayor capacidad se podría programar algoritmos más compactos que ahora mismo son inviables debido a su coste computacional.

Por último destacar que es conveniente mejorar en el algoritmo de clasificación de personas, ya que actualmente como se ha visto a lo largo de este proyecto, si se es muy estricto en este filtrado se dejan de detectar a algunas personas, pero si por el contrario se es poco estricto aparecen muchos falsos positivos.

Una vez indicadas las futuras líneas de investigación para mejorar el programa de detección de personas se procede a mencionar los posibles módulos que se le pueden añadir a esta aplicación para así obtener información más precisa.

Como se ve durante el desarrollo del proyecto, un problema importante reside en que cuando una persona permanece estática mucho tiempo en la escena deja de detectarse o se detecta de manera errónea. Es recomendable añadir un módulo que se encargue de solucionar este inconveniente pues evidentemente interesa seguir sabiendo que hay una persona en ese lugar.

Otro punto a mejorar es el seguimiento. Como se menciona desde que hay dos personas en la zona de trabajo este algoritmo es capaz de decir donde se encuentran pero en ningún momento es capaz de decir quien es cada cual. Esto es una información que puede ser muy interesante para un robot que quiere interactuar con las personas.

Por último es también muy interesante introducir un filtro predictor. Una vez que se sabe la posición de cada persona en una serie de frames sucesivos es muy interesante que un filtro determine la posible posición en el siguiente frame en caso de no tener información en esa imagen, bien sea debido a una gran cantidad de ruido exterior, o simplemente a que se produce un cruce de personas por lo que solo se detecta una persona.

Se desea que este proyecto pueda servir de ayuda a otras personas en futuros trabajos.

BIBLIOGRAFÍA

[1] Bradski,G.;Kaebler A. «*Learning OpenCV: Computer Vision with the OpenCV library*».

[2] Página web principal de las librerias OpenCV. «*http://opencv.willowgarage.com/wiki/*».

[3] Li,L.;Huang,W.;Yu-Hua Gu,I;Tian Q. «*Statical Modeling of Complex Background for Foreground Object Detection*». IEEE Trans. on Image Processing, 13 (11):1459-1472,2004.

[4] Chris Stauffer; Grimson,W.E.L. «*Adaptive background mixture models for real-time tracking*». Proc. of CVPR 1999, vol.2,pp.2246-2252.

[5] Beynon,M. «*Detecting packages in a multicamera video surveillance system*». Proc. of AVSS 2003, pp.221-228.

[6] Página web de introducción a las librerias OpenCV. «*http://futura.disca.upv.es/ imd/cursosAnteriors/2k3-2k4/copiaTreballs/serdelal/trabajoIMD.xml*».

[7] Chen,S; Xingzhi Luo;Bhandarkar, S.M. «*A multiscale parametric background Model for Stationary Foreground Object Detection*». Proc. of Motion and Video Computing 2007,8pp.

[8] De la Escalera,A. «*Visión por computador*». Prentice Hall.

[9] Martínez,J; Herrero,J;Orrite,C. «*Automatic Detection and Tracking using a Multi-camera UKF*». Proc. of PETS 2006, pp 59-66.

[10] Huwer,S;Niemann,H. «*Adaptative Change detection for Real-Time Surveillance Aplications*». Proc. of Visuals Surveillance 2000, pp.37-46.

[11] Guler,S;Farrow,K. «*People detection in crowded places*». Proc. of PETS 2006, June 18-23.

[12] Miezianko,R;Pokrajac,D. «*Detecting and Recognizing in Crowded Environments*». Proc. of Computer Vision System 2008, pp.241-250.

[13] Mathew, R.; Yu,Z.; Zhang,J. «*Detecting new stable objects in surveillance video*». Proc. of Multimedia Signal Processing 2005, pp. 1-4.

[14] Guler,S;Silverstein,J.A. «*Stacionary objects in multiple object tracking*». Proc. of AAVSS 2007, pp. 248-253.

[15] Porikli,F.;Ivanov,Y.;Haga,T. «*Robust Abandoned Object Detection Using Dual Foregrounds*». Journal on Advances in Signal Processing,art 30, 11pp.,2008.

[16] Rodríguez López,L. «*Análisis y evaluación de algoritmos de detección de movimiento*». 2009.

[17] Página web oficial de Ubuntu. «*http://www.ubuntu.com/*».

[18] Página web del profesor Antonio Sanz Montemayor. «*http://www.escet.urjc.es/ asanz/*».

[19] Li,L.; Leung.M. «*Integrating intensity and texture diferences for robust change detection*». IEEE Trans. on Image Processing, vol 11,pp.105-112,Feb 2002.

[20] Rosin,P. «*Tresholding for chance detection*». Proc.IEEE Int. Conf. Computer Vision, Jan 1998, pp.274-279.

[21] Li,L.;Huang,W.;Yu-Hua Gu,I;Tian Q. «*Foreground Object Detection from Videos Containing Complex Background*».

[22] Shih,M;Chang,Y;Fu B;Huang,C. «*Motion-based Background Modelind for Moving Object Detection on Moving Platforms*». Aug. 2007, pp. 1178-1182

[23] Grimsom,W;Stauffer,C;Romano,R. «*Using adaptative tracking to classify and monitor activities in a site*». IEEE Computer Society Conference Proceedings of Computer Vision And Pattern Recognitio, Jun 1998, pp.22-29.

[24] Zhang,W;Fang,X;Lin,W. «*Moving vehicles segmentation based on Gaussian motion mode*». Visual Communications and Image Processing 2005, vol 5960, 2005, pp.141-148.